MC68HC811E2

Technical Summary

8-Bit Microcontroller

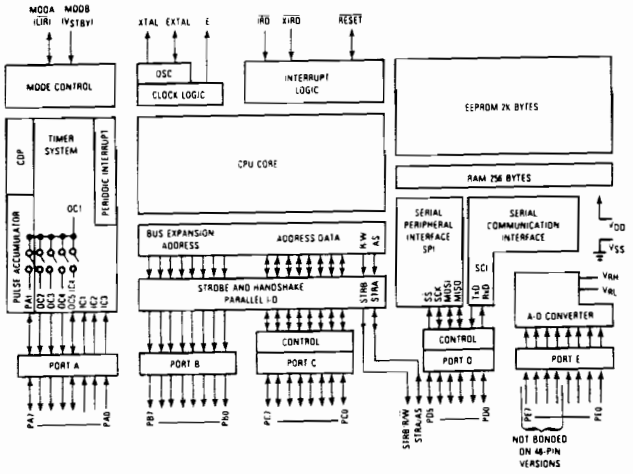
The MC68HC811E2 high density CMOS (HCMOS) microcontroller unit (MCU) contains highly sophisticated on-chip peripheral capabilities. This high-speed and low-power MCU has a nominal bus speed of two megahertz, and the fully static design allows operations at frequencies down to dc. This publication contains condensed information on the MCU; for detailed information, refer to *Advance Information Manual, HCMOS Single-Chip Microcontroller* (MC68HC11A8/D), *M68HC11 HCMOS Single-Chip Microcontroller Programmer's Reference Manual* (M68HC11RM/AD) or contact your local Motorola sales office.

Refer to the block diagram for the hardware features and to the list below for additional features available on the MCU.

- Enhanced 16-Bit Timer System with Four-Stage Programmable Prescaler
- Power Saving STOP and WAIT Modes
- Serial Peripheral Interface (SPI)
- Enhanced NRZ Serial Communications Interface (SCI)
- 8-Bit Pulse Accumulator Circuit
- Bit Test and Branch Instructions
- Real-Time Interrupt Circuit
- 2K Bytes of EEPROM
- 256 Bytes of Static RAM
- Eight-Channel 8-Bit A/D Converter

3

BLOCK DIAGRAM



This document contains information on a new product. Specifications and information herein are subject to change without notice.

OPERATING MODES

The MCU uses two dedicated pins (MODA and MODB) to select one of two basic operating modes or one of two special operating modes. The basic operating modes are single-chip and expanded-multiplexed; the special operating modes are bootstrap and special test. The following paragrphs describe the different modes.

SINGLE-CHIP MODE (MODE0)

In this mode, the MCU functions as a self-contained microcontroller and has no external address or data bus. This mode provides maximum use of the pins for on-chip peripheral functions, and all address and data activity occur within the MCU.

EXPANDED MULTIPLEXED MODE (MODE1)

In this mode, the MCU can address up to 64K bytes of address space. Higher-order address bits are output on the port B pins, and lower-order address bits and the data bus are mutliplexed on the port C pins. The AS pin provides the control output used in demultiplexing the low-order address at port C. The R/$\overline{W}$ pin is used to control the direction of data transfer on port C bus.

BOOTSTRAP MODE

In this mode, all vectors are fetched from the 192-byte on-chip bootloader ROM. This mode is very versatile and can be used for such functions as test and diagnostics on completed modules and for programming the EEPROM. The serial receive logic is initialized by software in the bootloader ROM, which provides program control for the serial communications interface (SCI) baud and word format. In this mode, a special control bit is configured that allows for self-testing of the MCU. This mode can be changed to other modes under program control.

TEST MODE

This mode is primarily intended for main production at time of manufacture; however, it may be used to program calibration or personality data into the internal EEPROM. In this mode, a special control bit is configured to permit access to a number of special test control bits. This mode can be changed to other modes under program control.

SIGNAL DESCRIPTION

V_{DD} AND V_{SS}

Power is supplied to the microcontroller using these two pins. V_{DD} is +5 volts (±0.5V) power, and V_{SS} is ground.

$\overline{RESET}$

This active low bidirectional control pin is used as an input to initialize the MCU to a known startup state and as an open-drain output to indicate that an internal failure has been detected in either the clock monitor or the computer operating properly (COP) circuit.

XTAL, EXTAL

These pins provide the interface for either a crystal or a CMOS-compatible clock to control the internal clock generator circuitry. The frequency applied shall be four times higher than the desired clock rate. Refer to Figure 1 for crystal and clock connections.

E

This pin provides an output for the internally generated E clock, which can be used for timing reference. The frequency of the E output is one-fourth that of the input frequency at the XTAL and EXTAL pins.

$\overline{IRQ}$

This pin provides the capability for asynchronously applying interrupts to the MCU. Either negative edge-sensitive or level-sensitive triggering is program selectable. This pin is configured to level-sensitive during reset. An external resistor connected to V_{DD} is required on $\overline{IRQ}$.

$\overline{XIRQ}$

This pin provides the capability for asynchronously applying non-maskable interrupts to the MCU after a power-on reset (POR). During reset, the X bit in the condition code register is set, and any interrupt is masked until enabled by software. This input is level-sensitive and requires an extenal pullup resistor to V_{DD}.

MODA/$\overline{LIR}$ AND MODB/V_{stby}

During reset, these pins are used to control the two basic operating modes and the two special operating modes. The $\overline{LIR}$ output can be used as an aid in debugging once reset is completed. The open-drain $\overline{LIR}$ pin goes to an active low during the first E-clock cycle of each instruction and remains low for the duration of that cycle. The V_{stby} (voltage standby) is used to retain RAM contents during device powerdown. The mode selections are shown below.

MODB	MODA	MODE SELECTED
1	0	Single Chip
1	1	Expanded Multiplexed
0	0	Special Bootstrap
0	1	Special Test

V_{RL} and V_{RH}

These pins provide the reference voltage for the A/D converter.

R/$\overline{W}$/STRB

This pin provides two different functions, depending on the operating mode. In single-chip mode, the pin provides STRB (output strobe) function; in the expanded-multiplexed mode, it provides R/$\overline{W}$ (read-write) function. The R/$\overline{W}$ is used to control the direction of transfers on the external data bus.

3

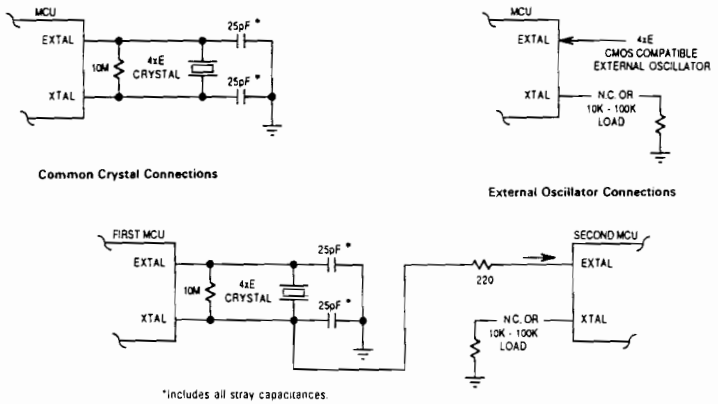


Figure 1. Oscillator Collections

AS/STRA

This pin provides two different functions depending on the operating mode. In single-chip mode, the pin provides STRA (input strobe) function, and in the expanded-multiplexed mode, it provides AS (address strobe) function. The AS may be used to demultiplex the address and data signals at port C.

INPUT/OUTPUT LINES (PA0-PA7, PB0-PB7, PC0-PC7, PD0-PD5, PE0-PE7)

These I/O lines are arranged into four 8-bit ports (A, B, C, and E) and one 6-bit port (D). All ports serve more than one purpose depending on the operating mode. Table 1 lists a summary of the pin functions to operating modes. Refer to **INPUT/OUTPUT PORTS** for additional information.

INPUT/OUTPUT PORTS

Port functions are controlled by the particular mode selected. In the single-chip mode and bootstrap mode, four ports are configured as parallel I/O data ports and port E can be used for general-purpose static inputs and/or analog-to-digital converter channel inputs. In the expanded-multiplexed mode and test mode, ports B, C, AS, and R/$\overline{W}$ are configured as a memory expansion bus. Table 1 lists the different port signals available. The following paragraphs describe each port.

PORT A

In all operating modes, port A may be configured for four input capture functions and three output compare functions; four output compare functions and three input capture functions; and a pulse a accumulator input (PAI) or a fifth output compare function. Each input capture pin provides for a transitional input, which is used to latch a timer value into the 16-bit input capture register. External devices provide the transitional inputs, and internal decoders determine which input transition edge is sensed. The output compare pins provide an output whenever a match is made between the value in the free-running counter (in the timer system) and a value loaded into the particular 16-bit output compare register. When port A bit 7 is configured as a PAI, the external input pulses are applied to the pulse accumulator system. The remaining port A lines may be used as general-purpose input or output lines.

PORT B

In the single-chip mode, all port B pins are general-purpose output pins. Port B may also be used in a simple strobed output mode where the STRB pulses each time port B is written. In the expanded-multiplexed mode, all

Table 1. Port Signal Functions

Port-Bit	Single-Chip and Bootstrap Mode	Expanded-Multiplexed and Special Test Mode
A-0	PA0/IC3	PA0/IC3
A-1	PA1/IC2	PA1/IC2
A-2	PA2/IC1	PA2/IC1
A-3	PA3/OC5/IC4/and-or OC1	PA3/OC5/IC4/and-or OC1
A-4	PA4/OC4/and-or OC1	PA4/OC4/and-or OC1
A-5	PA5/OC3/and-or OC1	PA5/OC3/and-or OC1
A-6	PA6/OC2/and-or OC1	PA6/OC2/and-or OC1
A-7	PA7/PAI/and-or OC1	PA7/PAI/and-or OC1
B-0	PB0	A8
B-1	PB1	A9
B-2	PB2	A10
B-3	PB3	A11
B-4	PB4	A12
B-5	PB5	A13
B-6	PB6	A14
B-7	PB7	A15
C-0	PC0	A0/D0
C-1	PC1	A1/D1
C-2	PC2	A2/D2
C-3	PC3	A3/D3
C-4	PC4	A4/D4
C-5	PC5	A5/D5
C-6	PC6	A6/D6
C-7	PC7	A7/D7
D-0	PD0/RxD	PD0/RxD
D-1	PD1/TxD	PD1/TxD
D-2	PD2/MISO	PD2/MISO
D-3	PD3/MOSI	PD3/MOSI
D-4	PD4/SCK	PD4/SCK
D-5	PD5/$\overline{SS}$	PD5/$\overline{SS}$
	STRA	AS
	STRB	R/$\overline{W}$
E-0	PE0/AN0	PE0/AN0
E-1	PE1/AN1	PE1/AN1
E-2	PE2/AN2	PE2/AN2
E-3	PE3/AN3	PE3/AN3
E-4	PE4/AN4##	PE4/AN4##
E-5	PE5/AN5##	PE5/AN5##
E-6	PE6/AN6##	PE6/AN6##
E-7	PE7/AN7##	PE7/AN7##

##Not Bonded in 48-Pin Versions

of the port B pins act as high-order (bits 8-15) address output pins.

PORT C

In the single-chip mode, port C pins are general-purpose input/output pins. Port C inputs can be latched by the STRA or may be used in full handshake modes of parallel I/O where the STRA input and STRB output acts as handshake control lines. In the expanded-multiplexed mode, port C pins are configured as multiplexed address/data pins. During the address cycle, bits 0 through 7 of the address are output on PC0-PC7; during the data cycle, bits 0 through 7 (PC0-PC7) are bidirectional data pins controlled by the R/$\overline{W}$ signal.

PORT D

In all modes, port D bits 0-5 may be used for general-purpose I/O or with the serial communications interface (SCI) and serial peripheral interface (SPI) subsystems. Bit 0 is the receive data input, and bit 1 is the transmit data output for the SCI. Bits 2 through 5 are used by the SPI subsystem.

PORT E

Port E is used for general-purpose static inputs and/or analog-to-digital channel inputs in all operating modes. Port E should not be read as static inputs while an A/D conversion is actually taking place.

MEMORY

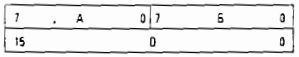
The memory maps for each mode of operation, a single-chip, expanded-multiplexed, special boot, and special test is shown in Figure 2. In the single-chip mode, the MCU does not generate external addresses. The internal memory locations are shown in the shaded areas, and the contents of the shaded areas are shown on the right side of the diagram. In the expanded-multiplexed mode, the memory locations are basically the same as the single-chip, except the memory locations between s shown in Figure 2. In the single-chip mode, the MCU does not (EXT) are for externally addressed memory and I O. The special bootstrap mode is similar to the single-chip mode, except the bootstrap program ROM is located at memory locations $BF40 through $BFFF. The special test mode is similar to the expanded-multiplexed mode, except the interrupt vectors are at external memory locations.

REGISTERS

The MCU contains the registers described in the following paragraphs.

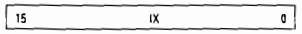
ACCUMULATOR A AND B

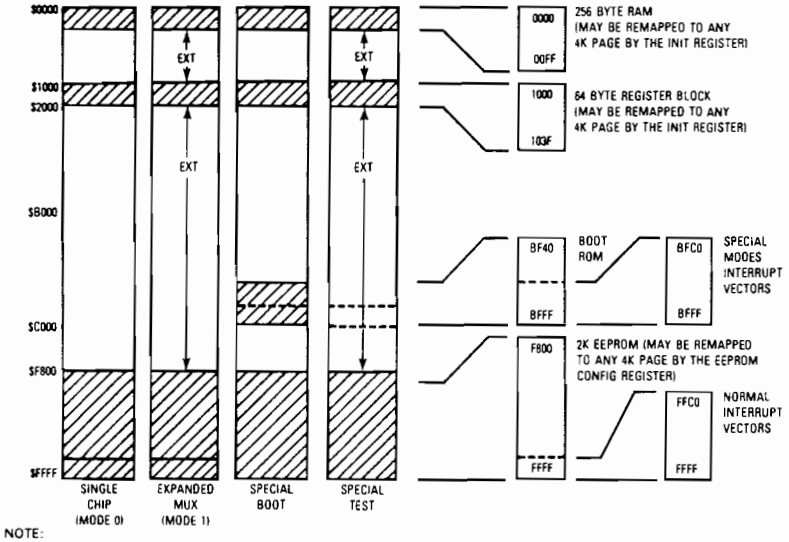
These accumulators are general-purpose 8-bit registers used to hold operands and results of arithmetic calculations or data manipulations. These two accumulators are treated as a single, double-byte accumulator called the D accumulator for some instructions.



INDEX REGISTER X (IX)

This index register is a 16-bit register used for the indexed addressing mode. It provides a 16-bit value that may be added to an 8-bit offset provided in an instruction to create an effective address. The index register may also be used either as a counter or a temporary storage area.





NOTE:
1. Either or both the internal RAM and registers can be remapped to any 4K boundary by software.

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
$1000	Bit 7	—	—	—	—	—	—	Bit 0	PORTA	I/O Port A
$1001									Reserved	
$1002	STAF	STAI	CWOM	HNDS	OIN	PLS	EGA	INVB	PIOC	Parallel I/O Control Register
$1003	Bit 7	—	—	—	—	—	—	Bit 0	PORTC	I/O Port C
$1004	Bit 7	—	—	—	—	—	—	Bit 0	PORTB	Output Port B
$1005	Bit 7	—	—	—	—	—	—	Bit 0	PORTCL	Alternate Latched Port C
$1006									Reserved	
$1007	Bit 7	—	—	—	—	—	—	Bit 0	DDRC	Data Direction for Port C
$1008			Bit 5	—	—	—	—	Bit 0	PORTD	I/O Port D
$1009			Bit 5	—	—	—	—	Bit 0	DDRD	Data Direction for Port D
$100A	Bit 7	—	—	—	—	—	—	Bit 0	PORTE	Input Port E
$100B	FOC1	FOC2	FOC3	FOC4	FOC5				CFORC	Compare Force Register
$100C	OC1M7	OC1M6	OC1M5	OC1M4	OC1M3				OC1M	OC1 Action Mask Register

Figure 2. Memory Map (Sheet 1 of 3)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
$1000	OC1D7	OC1D6	OC1D5	OC1D4	OC1D3				OC1D	OC1 Action Data Register
$100E	Bit 15	—	—	—	—	—	—	Bit 8	TCNT	Timer Counter Register
$100F	Bit 7	—	—	—	—	—	—	Bit 0		
$1010	Bit 15	—	—	—	—	—	—	Bit 8	TIC1	Input Capture 1 Register
$1011	Bit 7	—	—	—	—	—	—	Bit 0		
$1012	Bit 15	—	—	—	—	—	—	Bit 8	TIC2	Input Capture 2 Register
$1013	Bit 7	—	—	—	—	—	—	Bit 0		
$1014	Bit 15	—	—	—	—	—	—	Bit 8	TIC3	Input Capture 3 Register
$1015	Bit 7	—	—	—	—	—	—	Bit 0		
$1016	Bit 15	—	—	—	—	—	—	Bit 8	TOC1	Output Compare 1 Register
$1017	Bit 7	—	—	—	—	—	—	Bit 0		
$1018	Bit 15	—	—	—	—	—	—	Bit 8	TOC2	Output Compare 2 Register
$1019	Bit 7	—	—	—	—	—	—	Bit 0		
$101A	Bit 15	—	—	—	—	—	—	Bit 8	TOC3	Output Compare 3 Register
$101B	Bit 7	—	—	—	—	—	—	Bit 0		
$101C	Bit 15	—	—	—	—	—	—	Bit 8	TOC4	Output Compare 4 Register
$101D	Bit 7	—	—	—	—	—	—	Bit 0		
$101E	Bit 15	—	—	—	—	—	—	Bit 8	TI4O5	Output Compare 5 Register Output Compare 5 Register Input Capture 4 Register
$101F	Bit 7	—	—	—	—	—	—	Bit 0		
$1020	OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5	TCTL1	Timer Control Register 1
$1021	EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A	TCTL2	Timer Control Register 2
$1022	OC1I	OC2I	OC3I	OC4I	I4O5I	IC1I	IC2I	IC3I	TMSK1	Timer Interrupt Mask Reg. 1
$1023	OC1F	OC2F	OC3F	OC4F	I4O5F	IC1F	IC2F	IC3F	TFLG1	Timer Interrupt Flag Reg. 1
$1024	TOI	RTII	PAOVI	PAII			PR1	PR0	TMSK2	Timer Interrupt Mask Reg. 2
$1025	TOF	RTIF	PAOVF	PAIF					TFLG2	Timer Interrupt Flag Reg. 2
$1026	OORA7	PAEN	PAMOD	PEDGE	OORA3	I4/O5	RTR1	RTR0	PACTL	Pulse Accum. Control Reg.
$1027	Bit 7	—	—	—	—	—	—	Bit 0	PACNT	Pulse Accum. Count Reg.
$1028	SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0	SPCR	SPI Control Register
$1029	SPIF	WCOL		MODF					SPSR	SPI Status Register
$102A	Bit 7	—	—	—	—	—	—	Bit 0	SPDR	SPI Data Register
$102B	TCLR		SCP1	SCP0	RCKB	SCR2	SCR1	SCR0	BAUD	SCI Baud Rate Control
$102C	R8	T8		M	WAKE				SCCR1	SCI Control Register 1

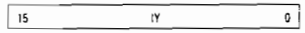
Figure 2. Memory Map (Sheet 2 of 3)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0		
$102D	TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK	SCCR2	SCI Control Register 2
$102E	TDRE	TC	RDRF	IDLE	OR	NF	FE		SCSR	SCI Status Register
$102F	Bit 7	—	—	—	—	—	—	Bit 0	SCDR	SCI Data (Read RDR, Write TDR)
$1030	CCF		SCAN	MULT	CD	CC	CB	CA	ADCTL	A/D Control Register
$1031	Bit 7	—	—	—	—	—	—	Bit 0	ADR1	A/D Result Register 1
$1032	Bit 7	—	—	—	—	—	—	Bit 0	ADR2	A/D Result Register 2
$1033	Bit 7	—	—	—	—	—	—	Bit 0	ADR3	A/D Result Register 3
$1034	Bit 7	—	—	—	—	—	—	Bit 0	ADR4	A/D Result Register 4
$1035				PTCON	BPRT3	BPRT2	BPRT1	BPRT0	BPROT	EEPROM Block Protect Reg.
$1036 Thru $1038									Reserved	
$1039	ADPU	CSEL	IRQE	DLY	CME		CR1	CR0	OPTION	System Configuration Options
$103A	Bit 7	—	—	—	—	—	—	Bit 0	COPRST	Arm/Reset COP Timer Cir.
$103B	ODD	EVEN		BYTE	ROW	ERASE	EELAT	EEPGM	PPROG	EEPROM Prog.Control Reg.
$103C	RBOOT	SMOD	MDA	IRV	PSEL3	PSEL2	PSEL1	PSEL0	HPRIO	Highest Priority I-Bit Int and Misc
$103D	RAM3	RAM2	RAM1	RAM0	REG3	REG2	REG1	REG0	INIT	RAM and I/O Mapping Reg.
$103E	TILOP		OCCR	CBYP	DISR	FCM	FCOP	TCON	TEST1	Factory TEST Control Register
$103F	EE3	EE2	EE1	EE0		NOCOP		EEON	CONFIG	COP, ROM, and EEPROM Enables

Figure 2. Memory Map (Sheet 3 of 3)

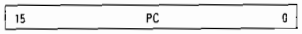
INDEX REGISTER Y (IY)

This index register is an 16-bit register used for the indexed addressing mode similar to the IX register; however, most instructions using the IY register are two-byte opcodes and require an extra byte of machine code and an extra cycle of execution time. The index register may also be used as a counter or a temporary storage area.



PROGRAM COUNTER (PC)

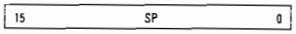
The program counter is a 16-bit register that contains the address of the next byte to be fetched.



STACK POINTER (SP)

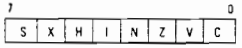
The stack pointer is a 16-bit register that contains the address of the next free location on the stack. The stack is configured as a sequence of last-in-first-out read/write registers, which allow important data to be stored during interrupts and subroutine calls. Each time a new byte is added to the stack, the SP is decremented; each time a byte is removed, the SP is incremented. The address contained in the SP also indicates the location at which the

accumulators A and B and registers IX and IY can be stored during certain instructions.



CONDITION CODE REGISTER (CCR)

The condition code register is an 8-bit register in which each bit is used to indicate the results of the instruction just executed. These bits can be individually tested by a program, and specific actions can be taken as a result of their state. Each bit is explained in the following paragraphs.



Carry/Borrow (C)

When set, this bit indicates that a carry or borrow out of the arithmetic logical unit (ALU) occurred during the last arithmetic operation. This bit is also affected during shift and rotate instructions.

Overflow (V)

The overflow bit is set if an arithmetic overflow occurred as a result of the operation; otherwise, the V bit is cleared.

Zero (Z)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was zero.

Negative (N)

When set, this bit indicates that the result of the last arithmetic, logical, or data manipulation was negative (the MSB of the result is a logic one).

Interrupt (I)

This bit is set either by hardware or program instruction to disable (mask) all maskable interrupt sources (both external and internal).

Half Carry (H)

This bit is set during ADD, ABA, and ADC operations to indicate that a carry occurred between bits 3 and 4. This bit is mainly useful in BCD calculations.

X Interrupt Mask (X)

This mask bit is set only by hardware (reset or $\overline{XIRQ}$) and is cleared only by program instruction (TAP or RTI).

Stop Disable (S)

This bit, under program control, is set to disable the STOP instruction, and is cleared to enable the STOP instruction. The STOP instruction is treated as no operation (NOP) if the S bit is set.

RESETS

The MCU can be reset four ways: 1) an active low input to the $\overline{RESET}$ pin; 2) a power-on reset function; 3) a computer operating properly (COP) watchdog-timer timeout; and 4) a clock monitor failure. The $\overline{RESET}$ input consists mainly of a Schmitt trigger that senses the $\overline{RESET}$ line logic level.

RESET PIN

To request an external reset, the $\overline{RESET}$ pin must be held low for eight E_{CYC} (two E_{CYC} if no distinction is needed between internal and external resets). To prevent the EEPROM contents from being corrupted during power transitions, the reset line should be held low while V_{DD} is below its minimum operating level. A low voltage inhibit (LVI) circuit is required to protect EEPROM from corruption as shown in Figure 3.

POWER-ON RESET (POR)

Power-on reset occurs when a positive transition is detected on V_{DD}. The power-on reset is used strictly for power turn-on conditions and should not be used to detect any drop in the power supply voltage. If the external $\overline{RESET}$ pin is low at the end of the power-on delay time, the processor remains in the reset condition until $\overline{RESET}$ goes high.

COMPUTER OPERATING PROPERLY (COP) RESET

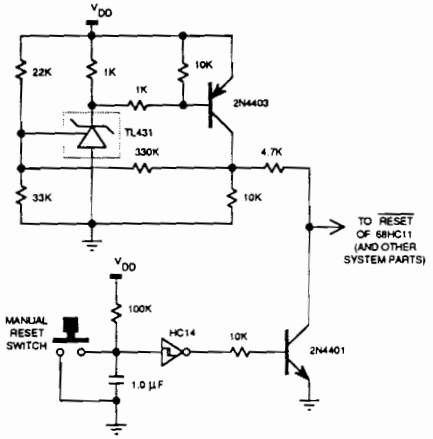
The MCU contains a watchdog timer that automatically times out if not reset within a specific time by a program reset sequence. If the COP watchdog timer is allowed to timeout, a reset is generated, which drives the $\overline{RESET}$ pin low to reset the MCU and the external system.

The COP reset function can be enabled or disabled by setting the control bit in an EEPROM cell of the system configuration register. Once programmed, this control bit remains set (or cleared) even when no power is applied, and the COP function is enabled or disabled independent of resident software. Protected control bits (CR1 and CR0), in the configuration options register, allow the user to select one of four COP timeout rates. Table 2 shows the relationship between CR1 and CR0 and the COP timeout period for various system clock frequencies.

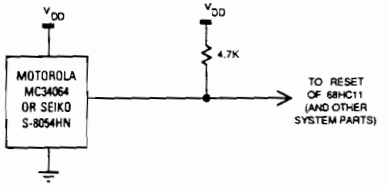
CLOCK MONITOR RESET

The MCU contains a clock monitor circuit which measures the E clock input frequency. If the E clock input rate is above 200 kHz, then the clock monitor does not generate a MCU reset. If the E clock signal is lost or its frequency falls below 10 kHz, then a MCU reset is generated, and the $\overline{RESET}$ pin is driven low to reset the external system.

The clock monitor reset can be enabled or disabled by a read-write control bit (CME) in the system configuration options register.



Reset Circuit with LVI and RC Delay



Simple LVI Reset Circuit

Figure 3. Typical LVI Reset Circuits

Table 2. COP Timeout Periods

CR1	CR0	E/2^{15} Divided By	XTAL = 2^{23} Timeout −1/+15.6 ms	XTAL = 8.0 MHz Timeout −0/+16.4 ms	XTAL = 4.9152 MHz Timeout −0/+26.7 ms	XTAL = 4.0 MHz Timeout −0/+32.8 ms	XTAL = 3.6864 MHz Timeout −0/+35.6 ms
0	0	1	15.625 ms	16.384 ms	26.667 ms	32.768 ms	35.556 ms
0	1	4	62.5 ms	65.536 ms	106.67 ms	131.07 ms	142.22 ms
1	0	16	250 ms	262.14 ms	426.67 ms	524.29 ms	568.89 ms
1	1	64	1 s	1.049 s	1.707 s	2.1 s	2.276 s
		E =	2.1 MHz	2.0 MHz	1.2288 MHz	1.0 MHz	921.6 kHz

INTERRUPTS

There are seventeen hardware and one software interrupts (excluding reset type interrupts) that can be generated from all the possible sources. These interrupts can be divided into two categories, maskable and non-maskable. Fifteen of the interrupts can be masked with the condition code register I bit. All the on-chip interrupts are individually maskable by local control bits. The software interrupt is non-maskable. The external input to the $\overline{XIRQ}$ pin is considered a non-maskable interrupt because, once enabled, it cannot be masked by software; however, it is masked during reset and upon receipt of an interrupt at the $\overline{XIRQ}$ pin. The last interrupt, illegal opcode, is also a non-maskable interrupt. Table 3 provides a list of each interrupt, its vector location in ROM, and the actual condition code and control bits that mask it. Figure 4 shows the interrupt stacking order.

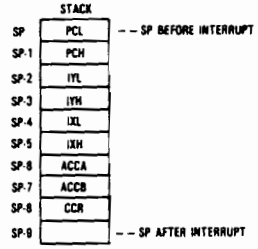


Figure 4. Stacking Order

SOFTWARE INTERRUPT (SWI)

The SWI is executed the same as any other instruction and will take precedence over interrupts only if the other interrupts are masked (I and X bits in the CCR set). The SWI execution is similar to the maskable interrupts such as setting the I bit, CPU registers are stacked, etc.

NOTE

The SWI instruction cannot be fetched as long as another interrupt is pending execution. However, once fetched, no other interrupt can be honored until the first instruction in the SWI service routine is completed.

Table 3. Interrupt Vector Assignments

Vector Address	Interrupt Source	CC Register Mask	Local Mask
FFC0, C1	Reserved	—	—
•	•		
•	•		
FFD4, D5,	Reserved	—	—
FFD6, D7	SCI Serial System	I Bit	
	Receive Data Register Full		RIE
	Receive Overrun		RIE
	Idle Line Detect		ILIE
	Transmit Data Register Empty		TIE
	Transmit Complete		TCIE
FFD8, D9	SPI Serial Transfer Complete	I Bit	SPIE
FFDA, DB	Pulse Accumulator Input Edge	I Bit	PAII
FFDC, DD	Pulse Accumulator Overflow	I Bit	PAOVI
FFDE, DF	Timer Overflow	I Bit	TOI
FFE0, E1	Timer Input Capture 4/Output Compare 5	I Bit	I4O5I
FFE2, E3	Timer Output Compare 4	I Bit	OC4I
FFE4, E5	Timer Output Compare 3	I Bit	OC3I
FFE6, E7	Timer Output Compare 2	I Bit	OC2I
FFE8, E9	Timer Output Compare 1	I Bit	OC1I
FFEA, EB	Timer Input Capture 3	I Bit	OC3I
FFEC, ED	Timer Input Capture 2	I Bit	OC2I
FFEE, EF	Timer Input Capture 1	I Bit	OC1I
FFF0, F1	Real-Time Interrupt	I Bit	RTII
FFF2, F3	$\overline{IRQ}$ (External Pin or Parallel I/O)	I Bit	
	External Pin		None
	Parallel I/O Handshake		STAI
FFF4, F5	$\overline{XIRQ}$ Pin (Pseudo Non-Maskable Interrupt)	X Bit	None
FFF6, F7	SWI	None	None
FFF8, F9	Illegal Opcode Trap	None	None
FFFA, FB	COP Failure (Reset)	None	NOCOP
FFFC, FD	COP Clock Monitor Fail (Reset)	None	CME
FFFE, FF	$\overline{RESET}$	None	None

ILLEGAL OPCODE TRAP

Since not all possible opcodes or opcode sequences are defined, an illegal opcode detection circuit has been included in the MCU. When an illegal opcode is detected, an interrupt is requested to the illegal opcode vector.

REAL-TIME INTERRUPT

The real-time interrupt provides a programmable periodic interrupt. This interrupt is maskable by either the I bit in the CCR or the RTII control bit. The rate is based on the MCU E clock and is software selectable to be E/2^{13}, E/2^{14}, E/2^{15}, or E/2^{16}.

LOW-POWER MODES

The MCU contains two programmable low-power operating modes: stop and wait. In the wait mode, the on-chip oscillator remains active; in the stop mode, the oscillator is stopped. The following paragraphs describe the two low-power modes.

STOP

The STOP instruction places the MCU in its lowest power consumption mode, provided the S bit in the CCR is clear. In this mode, all clocks are stopped, thereby halting all internal processing.

To exit the stop mode, a low level must be applied to either $\overline{IRQ}$, $\overline{XIRQ}$ or $\overline{RESET}$. An external interrupt used at $\overline{IRQ}$ is only effective if the I bit in the CCR is clear. An external interrupt applied at the $\overline{XIRQ}$ input would be effective regardless of the X-bit setting in the CCR; however, the actual recovery sequence differs, depending on the X-bit setting. If the X bit is clear, the MCU starts with the stacking sequence leading to the normal service of the $\overline{XIRQ}$ request. If the X bit is set, the processing will always continue with the instruction immediately following the STOP instruction. A low input to the $\overline{RESET}$ pin will always result in an exit from the stop mode, and the start of MCU operations is determined by the reset vector.

A restart delay is required if the internal oscillator is being used, to allow the oscillator to stabilize when exiting the stop mode. If a stable external oscillator is being used, a control bit in the OPTION register may be used (cleared) to bypass the delay. If the control bit is clear, then the $\overline{RESET}$ pin would not normally be used for exiting the stop mode. In this case, the reset sequence sets the delay control bit and the restart delay will be imposed.

WAIT

The wait (WAI) instruction places the MCU in a low-power consumption mode, but the wait mode consumes slightly more power than the stop mode. In the wait mode, the oscillator is kept running. Upon execution of the WAIT instruction, the machine state is stacked and program execution stops. The wait state can only be exited by an unmasked interrupt or $\overline{RESET}$. If the I bit is set and the COP is disabled, the timer system will be turned off to further reduce power consumption. The amount of power savings is application dependent and depends upon circuitry connected to the MCU pins and upon subsystems (i.e., timer, SPI, SCI) that are active when the wait mode is entered. Turning off the A/D subsystem by clearing ADPU further reduces wait mode current.

PROGRAMMABLE TIMER

The timer system uses a "time-of-day" approach in that all timing functions are related to a single 16-bit free-running counter. The free-running counter is clocked by the output of a programmable prescaler (divide by 1, 4, 8, or 16), which is, in turn, clocked by the MCU E clock. The free-running counter can be read by software at any time without affecting its value because it is clocked and read on opposite half cycles of the E clock. The counter is cleared on reset and is a read-only register. The counter repeats every 65,536 counts, and when the count changes from $FFFF to $0000, a timer overflow flag bit is set. The overflow flag also generates an internal interrupt if the overflow interrupt enable bit is set. The timer has four input capture and five output compare functions. The functions and registers of the timer are explained in the following paragraphs.

INPUT CAPTURE FUNCTION

There are four 16-bit read-only input capture registers that are not affected by reset. Each register is used to latch the value of the free-running counter when a selected transition at an extenal pin is detected. External devices provide the inputs on the PA0-PA3 pins, and an interrupt can be generated when an input capture edge is detected. The time of detection can be read from the appropriate register as part of the interrupt routine.

Port A pin 3 serves multiple functions. After reset, data direction bit 3 (DDRA3), in the PACTL register is cleared to zero configuring port A pin 3 as an input. Port A pin 3 can then be used as a input capture 4 (IC4), by setting I4/O5 to "one" in the PACTL register. The I4/O5 bit is configured to OC5 (cleared to zero) on reset. If DDRA3 is configured as an output and IC4 is enabled, writes to port A bit 3 causes edges on the PA3 to result in input captures. When the TI405 register is acting as the IC4 capture register it cannot be written to. When PA3 is being used as IC4, writes to TI405 register have no meaning.

TIMER CONTROL REGISTER 2 (TCTL2) $1021

7	6	5	4	3	2	1	0
EDG4B	EDG4A	EDG1B	EDG1A	EDG2B	EDG2A	EDG3B	EDG3A
RESET: 0	0	0	0	0	0	0	0

EDGxB and EDGxA — Input Capture x Edge Control
These two bits (EDGxB and EDGxA) are cleared to zero by reset and are encoded to configure the input sensing logic for input capture x.

EDGxB	EDGxA	Configuration
0	0	Capture disabled
0	1	Capture on rising edges only
1	0	Capture on falling edges only
1	1	Capture on any (rising or falling) edge

OUTPUT COMPARE FUNCTION

There are five 16-bit read/write output compare registers, which are set to $FFFF on reset. A value written into the SE registers is compared to the free-running counter value during each E-clock cycle. If a match is found, the particular output compare flag is set, and an interrupt is generated, provided that particular interrupt is enabled.

In addition to the interrupt, a specified action may be initiated at a timer output pin(s). For output compare one (OC1), the output action to be taken when a match is found is controlled by a 5-bit mask register and a 5-bit data register. The mask register specifies which timer port outputs are to be used, and the data register specifies what data is placed on the SE timer ports. For OC2 through OC5, one specific timer output is affected as controlled by the two-bit fields in a timer control register. These actions include: 1) timer disconnect from output pin logic, 2) toggle output compare line, 3) clear output compare line to zero, or 4) set output compare line to one. Upon reset, I4/O5 is configured as OC5. The OC5 function overrides DDRA3 to force the Port A pin 3 to be an output whenever OM5:OL5 bits are not 0:0. In all other aspects, OC5 works the same as the other output compares.

TIMER COMPARE FORCE REGISTER (CFORC) $100B

This 8-bit write-only register is used to force early output compare actions. This compare force function is not recommended for use with the output toggle function because a normal compare occurring immediately before or after the force may result in undersirable operation.

7	6	5	4	3	2	1	0
FOC1	FOC2	FOC3	FOC4	FOC5	0	0	0
RESET: 0	0	0	0	0	0	0	0

FOC1-FOC5 — Force Output Compare x Action
1 = Causes action progrmmed for output compare x, except the OCxF flag bit is not set
0 = Has no meaning
Bits 2-0 — Not Implemented
These bits always read zero.

OUTPUT COMPARE 1 MASK REGISTER (OC1M) $100C

This register is used with output compare 1 to specify the bits of port A which are affected as a result of a successful OC1 compare.

7	6	5	4	3	2	1	0
OC1M7	OC1M6	OC1M5	OC1M4	OC1M3	0	0	0
RESET: 0	0	0	0	0	0	0	0

Set bit(s) to enable OC1 to control corresponding pin(s) of port A.

OUTPUT COMPARE 1 DATA REGISTER (OC1D) $100D

This register is used with output compare 1 to specify the data which is to be stored to the affected bit of port A as a result of a successful OC1 compare.

7	6	5	4	3	2	1	0
OC1D7	OC1D6	OC1D5	OC1D4	OC1D3	0	0	0
RESET: 0	0	0	0	0	0	0	0

If OC1Mx is set, data in OC1Dx is output to port A bit-x on successful OC1 compares.

TIMER CONTROL REGISTER (TCTL1) $1020

7	6	5	4	3	2	1	0
OM2	OL2	OM3	OL3	OM4	OL4	OM5	OL5
RESET: 0	0	0	0	0	0	0	0

OM2-OM5 — Output Mode
OL2-OL5 — Output Level
These control bit pairs (OMx and OLx) are encoded to specify the output action taken as a result of a successful OCx compare.

OMx	OLx	Action Taken Upon Successful Compare
0	0	Timer disconnected from output pin logic
0	1	Toggle OCx output line
1	0	Clear OCx output line to zero
1	1	Set OCx output line to one

TIMER INTERRUPT MASK REGISTER 1 (TMSK1)

7	6	5	4	3	2	1	0
OC1I	OC2I	OC3I	OC4I	I4O5I	IC1I	IC2I	IC3I
RESET: 0	0	0	0	0	0	0	0

OCxI — Output Compare x Interrupt
1 = Interrupt sequence requested if OCxF = 1 in TFLG1
0 = Interrupt inhibited
ICxI — Input Capture x Interrupt
1 = Interrupt sequence requested if ICxF = 1 in TFLG1
0 = Interrupt inhibited

NOTE

When the I4/O5 bit in the PACTL register is one, the I4O5I bit behaves as the input capture 4 interrupt bit. When I4/O5 is zero, the I4O5I bit acts as the output compare 5 interrupt control bit.

TIMER INTERRUPT FLAG REGISTER 1 (TFLG1)

This register is used to indicate the occurrence of timer system events and, with the TMSK1 register, allows the timer subsystem to operate in a polled or interrupt driven system. Each bit in the TFLG1 has a corresponding bit in the TMSK1 in the same bit position.

7	6	5	4	3	2	1	0
OC1F	OC2F	OC3F	OC4F	I4O5F	IC1F	IC2F	IC3F
RESET: 0	0	0	0	0	0	0	0

OCxF — Output Compare x Flag
Set each time the timer counter matches the output compare register x value. To clear a flag bit in TFLG1, you must write a "one" to the corresponding bit position(s).
1 = Bit cleared
0 = Not affected

ICxF — Input Capture x Flag
Set each time a selected active edge is detected on the ICx input line. To clear a flag bit in TFLG1, you must write a "one" to the corresponding bit position(s).
1 = Bit cleared
0 = Not affected

NOTE

When the I4/O5 bit in the PACTL register is one, the I4O5F bit behaves as the input capture 4 flag bit. When I4/O5 is zero, the I4O5I bit acts as the output compare 5 flag.

TIMER INTERRUPT MASK REGISTER 2 (TMSK2) $1024

This register is used to control whether or not a hardware interrupt sequence is requested as a result of a status bit being set in TFLG1. Two timer prescaler bits are also included in this register.

7	6	5	4	3	2	1	0
TOI	RTII	PAOVI	PAII	0	0	PR1	PR0
RESET: 0	0	0	0	0	0	0	0

TOI — Timer Overflow Interrupt Enable
1 = Interrupt request when TOF = 1
0 = TOF interrupt disabled

RTII — RTI Interrupt Enable
1 = Interrupt requested when RTIF = 1
0 = RTIF interrupt disabled

PAOVI — Pulse Accumulator Overflow interrupt Enable
1 = Interrupt requested when PAOVF = 1
0 = PAOVF disabled

PAII — Pulse Accumulator Input Interrupt Enable
1 = Interrupt requested when PAIF = 1
0 = PAIF disabled

Bits 3-2 — Not Implemented
These bits always read zero.

PR1 and PR0 — Timer Prescaler Selects
Can only be written to during initialization. Writes are disabled after the first write or after 64 E cycles out of reset.

PR1	PR0	Divide-by-Factor
0	0	1
0	1	4
1	0	8
1	1	16

TIMER INTERRUPT FLAG REGISTER 2 (TFLG2) $1025

This register is used to indicate the occurrence of timer system events and, with the TMSK2 register, allows the timer subsystem to operate in a polled or interrupt driven system. Each bit in the TFLG2 has a corresponding bit in the TMSK2 in the same bit position.

7	6	5	4	3	2	1	0
TOF	RTIF	PAOVF	PAIF	0	0	0	0
RESET: 0	0	0	0	0	0	0	0

TOF — Timer Overflow
Set to one each time the 16-bit free-running counter advances from a value of $FFFF to $0000. Cleared by a write to TFLG2 with bit 7 set.

RTIF — Real-Time Interrupt Flag
Set at each rising edge of the selected tap point. Cleared by a write to TFLG2 with bit 6 set.

PAOVF — Pulse-Accumulator Overflow Interrupt Flag
Set when the count in the pulse accumulator rolls over from $FF to $00. Cleared by a write to the TFLG2 with bit 5 set.

PAIF — Pulse-Accumulator Input-Edge Interrupt Flag
Set when an active edge is detected on the PAI input pin. Cleared by a write to TFLG2 with bit 4 set.

Bits 3-0 — Not Implemented
These bits always read zero.

PULSE ACCUMULATOR

The pulse accumulator is an 8-bit counter that can operate in either of two modes, depending on the state of a control bit in the PACTL register. These are the event counting mode and the gated time accumulation mode. In the event counting mode, the 8-bit counter is clocked to increasing values by an external pin. The maximum clocking rate for the external event counting mode is E clock divided by two. In the gated time accumulation mode, a free-running E clock/64 signal drives the 8-bit counter, but only while the external PAI input pin is activated.

PULSE ACCUMULATOR CONTROL REGISTER (PACTL) $1026

Four bits in this register are used to control an 8-bit pulse accumulator system, and two other bits are used to select the rate for the real-time interrupt system.

7	6	5	4	3	2	1	0
DDRA7	PAEN	PAMOD	PEDGE	DDRA3	I4/O5	RTR1	RTR0
RESET: 0	0	0	0	0	0	0	0

DDRA7 — Data Direction for Port A Bit 7
1 = Output
0 = Input only

PAEN — Pulse-Accumulator System Enable
1 = Pulse accumulator on
0 = Pulse accumulator off

PAMOD — Pulse Accumulator Mode
1 = Gated time accumulator
0 = External even counting

RTR1	RTR0	Divide E By	XTAL = 2^{23}	XTAL = 8.0 MHz	XTAL = 4.9152 MHz	XTAL = 4.0 MHz	XTAL = 3.6864 MHz
0	0	2^{13}	3.91 ms	4.10 ms	6.67 ms	8.19 ms	8.89 ms
0	1	2^{14}	7.81 ms	8.19 ms	13.33 ms	16.38 ms	17.78 ms
1	0	2^{15}	15.62 ms	16.38 ms	26.67 ms	32.77 ms	35.56 ms
1	1	2^{16}	31.25 ms	32.77 ms	53.33 ms	65.54 ms	71.11 ms
		E =	2.1 MHz	2.0 MHz	1.2288 MHz	1.0 MHz	921.6 kHz

PEDGE — Pulse Accumulator Edge Control
This bit provides clock action along with PAMOD.
1 = Sensitive to rising edges at PAI pin if PAMOD = 0. In gated accumulation mode counting is enabled by a low on PAI pin if PAMOD = 1.
0 = Sensitive to falling edges at PAI pin if PAMOD = 0. In gated accumulation mode counting is enabled by a high on PAI pin if PAMOD = 1.

DDRA3 — Data Directional for Port A Bit 3
1 = Output
0 = Input only

I4/O5 — Input 4/Output 5
1 = Input capture 4 function enabled (No OC5)
0 = Output compare 5 function enabled (No IC4)

RTR1 and RTR0 — RTI Interrupt Rate Selects
These two bits select one of four rates for the real-time periodic interrupt circuits. Reset clears these two bits and after reset, a full RTI period elapses before the first RTI interrupt.

EEPROM PROGRAMMING

The 2K bytes of EEPROM are located at $F800 through $FFC0. Programming of the EEPROM is controlled by the EEPROM programming control register (PPROG). The EEPROM is disabled when the EEON bit in the system configuration register (CONFIG) is zero. Programming and erasure of the EEPROM relies on an internal high-voltage charge pump. At E clock frequencies below 2 MHz, the efficiency of this charge pump decreases, which increases the time required to program or erase a location. Recommended program and erase time is 10 milliseconds when the E clock is between 2 MHz and should be increased to as much as 20 milliseconds when E clock is between 1 MHz and 2 MHz. When E clock is below 1 MHz, the clock source for the charge pump should be switched from the system clock to an on-chip R-C oscillator clock. This is done by setting the CSEL bit in the OPTION register. A 10 millisecond period should be allowed after setting the CSEL bit to allow the charge pump to stabilize. The following paragraphs describe how to program or erase the EEPROM using the PPROG control register.

EEPROM BLOCK PROTECT REGISTER (BPROT) $1035

This 5-bit register protects against inadvertent writes to the CONFIG register and to the EEPROM. To permit the user to separate EEPROM into categories like 'temporary' or 'permanent', EEPROM is divided into four individually protected blocks. The CONFIG register is also protected.

In normal operating modes, EEPROM and CONFIG are protected out of reset, and the user has 64 E clock cycles to unprotect any of the blocks that will require programming or erasing. The BPROT register bits can only be cleared, written to zero, during the first 64 E clock cycles after reset. Once the bits are cleared, the associated EEPROM section and/or the CONFIG register can be programmed or erased in the normal manner. The EEPROM is visible only if the EEON bit in the CONFIG register is set. In the test or bootstrap modes, bits of the BPROT register can be set or cleared at any time. In either single-chip or expanded mode, BPROT register bits can be written back to one anytime after the first 64 E clock cycles in order to protect the EEPROM and/or the CONFIG register. However, these bits can only be cleared again in the test or bootstrap modes.

7	6	5	4	3	2	1	0
0	0	0	PTCON	BPRT3	BPRT2	BPRT1	BPRT0
RESET: 0	0	0	1	1	1	1	1

Bits 7-5 — Not Implemented
These bits always read zero

PTCON — Protect CONFIG Register Bit
1 = Programming/erasure of the CONFIG register disabled
0 = Programming/erasure of the CONFIG register allowed

BPRT3–BPRT0 — Block Protect Bits
1 = A set bit protects a block of EEPROM against programming or erasure.
0 = A cleared bit permits programming or erasure of the associated block.

Bit	Block Protected	Block Size
BPRT0	$1800–19FF	512 Bytes
BPRT1	$1A00–1BFF	512 Bytes
BPRT2	$1C00–1DFF	512 Bytes
BPRT3	$1E00–1FFF	512 Bytes

ERASING THE EEPROM

Erasure of the EEPROM is controled by bit settings in PPROG, and the appropriate bits in the BPROT register must also be cleared before the EEPROM can be changed. Programs can be written to perform bulk, row, or byte erase. In bulk erase, all 512 bytes of the EEPROM are erased. In row erase, 16 bytes ($B600-$B60F, $B610-$B61F), etc) are erased. Other MCU operations can continue to be performed during erasing provided the operations do not include reads of data from EEPROM.

PROGRAMMING EEPROM

During programming, the ROW and BYTE bits are not used. If the E clock frequency is 1 MHz or less, the CSEL bit in the OPTION register must be set. Zeros must be erased by a separate erase operation before programming. Other MCU operations can continue to be performed during programming provided the operations do not include reads of data from EEPROM.

EEPROM PROGRAMMING CONTROL REGISTER (PPROG) $103B

This 8-bit register is used to control programming and erasure of the EEPROM. This register is cleared on reset so the EEPROM is configured for normal reads.

7	6	5	4	3	2	1	0
ODD	EVEN	0	BYTE	ROW	ERASE	EELAT	EEPGM
RESET: 0	0	0	0	0	0	0	0

ODD — Program Odd Rows (TEST)
EVEN — Program Even Rows (TEST)
Bit 5 — Not Implemented
This bit always reads zero.
BYTE — Byte Erase Select
This bit overrides the ROW bit.
1 = Erase only one byte
0 = Row or bulk erase
ROW — Row Erase Select
If BYTE bit = 1, ROW has no meaning.
1 = Row erase
0 = Bulk or byte erase
ERASE — Erase Mode Select
1 = Erase mode
0 = Normal read or program
EELAT — EEPROM Latch Control
1 = EEPROM Address and data configured for programming/erasing
0 = EEPROM Address and data configured for read mode
EEPGM — EEPROM Programming Voltage Enable
1 = Programming voltage turned on
0 = Programming voltage turned off

NOTE

A strict register access sequence must be followed to allow successful programming and erase operations. The following procedures for modifying the EEPROM and CONFIG register detail the sequence. If an attempt is made to set both the EELAT and EEPGM bits in the same write cycle and if this attempt occurs before the required write cycle with the EELAT bit set, then neither is set. If a write to an EEPROM address is performed while the EEPGM bit is set, the write is ignored, and the programming operation in progress is not disturbed. If no EEPROM address is written between when EELAT is set and EEPGM is set, then no program or erase operation takes place. These safeguards were included to prevent accidental EEPROM changes in cases of program runaway.

ERASING THE CONFIG REGISTER

Erasing the CONFIG register follows the same procedures as that used for the EEPROM including bulk, byte, and row erase. The CONFIG register may be programmed or erased while the MCU is operating in any mode depending on the setting of bit A in BPROT. The bulk erase restriction on CONFIG is not present on all derivatives in the M68HC11 Family. Please check the applicable data sheet or technical summary for the restrictions.

PROGRAMMING THE CONFIG REGISTER

Programming the CONFIG register follows the same procedures as that used for the EEPROM except the CONFIG register address is used. On mask set B96D, the CONFIG register may only be programmed while the MCU is operating in the test or bootstrap mode.

SYSTEM CONFIGURATION REGISTER (CONFIG) $103F

The CONFIG is implemented in EEPROM cells and controls the presence of ROM and EEPROM in the memory map and enables the COP watchdog system.

7	6	5	4	3	2	1	0
EE3	EE2	EE1	EE0	0	NOCOP	0	EEON

EE0-EE3 — EEPROM Map Position
These four bits specify the upper four bits of the EEPROM address. These bit have no meaning in the single-chip mode, because the 2K EEPROM is forced on at locations $F800 through $FFFF.

EE3	EE2	EE1	EE0	Location
0	0	0	0	$0800–$0FFF
0	0	0	1	$1800–$1FFF
0	0	1	0	$2800–$2FFF
0	0	1	1	$3800–$3FFF
0	1	0	0	$4800–$4FFF
0	1	0	1	$5800–$5FFF
0	1	1	0	$6800–$6FFF
0	1	1	1	$7800–$7FFF
1	0	0	0	$8800–$8FFF
1	0	0	1	$9800–$9FFF
1	0	1	0	$A800–$AFFF
1	0	1	1	$B800–$BFFF
1	1	0	0	$C800–$CFFF
1	1	0	1	$D800–$DFFF
1	1	1	0	$E800–$EFFF
1	1	1	1	$F800–$FFFF

Bit 3 — Not Implemented
This bit always reads zero
NOCOP — COP System Disable
1 = COP watchdog system disable
0 = COP watchdog system enabled
Bit 1 — Not Implemented
This bit always reads zero
EEON — Enable On-Chip EEPROM
When this bit is programmed to "zero", the 512-byte EEPROM is disabled, and that memory space becomes externally accessed space.

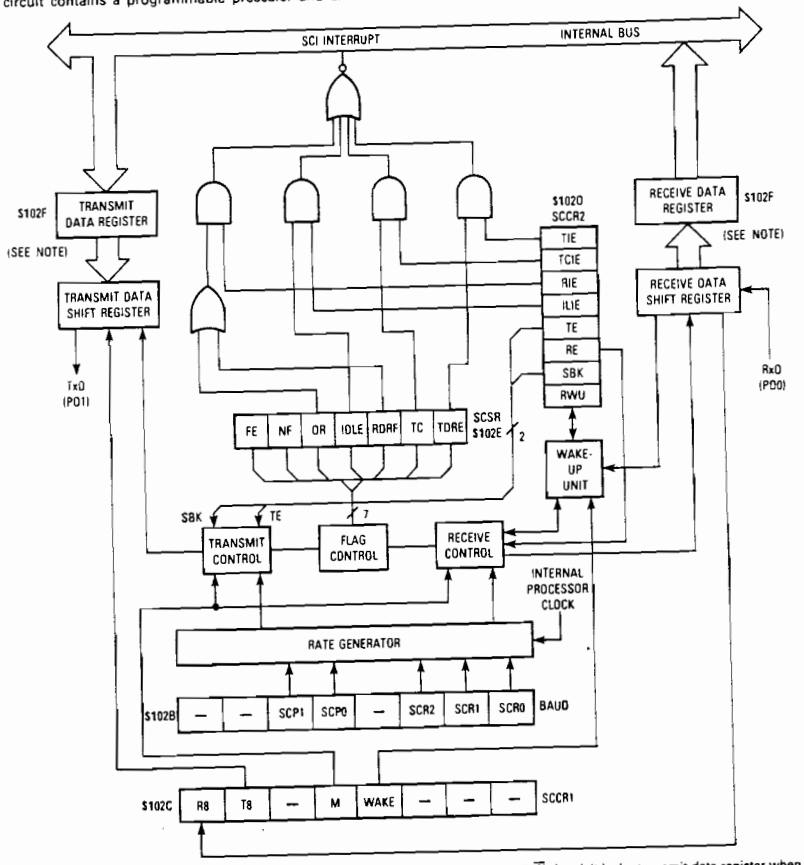
SERIAL COMMUNICATIONS INTERFACE

The serial communications interface (SCI) allows the MCU to be interfaced efficiently with peripheral devices that require an asynchronous serial data format. The SCI uses a standard NRZ format with a variety of baud rates derived from the crystal clock circuit. Interfacing is accomplished using port D pins PD0 for receive data (RxD), and PD1 for transmit data (TxD). The baud rate generation circuit contains a programmable prescaler and divider clocked by the MCU E clock. Figure 5 shows a block diagram of the SCI.

DATA FORMAT

Receive data in or transmit data out is the serial data presented between the PD0 and the internal data bus and between the internal data bus and PD1. The data format requires

1) An idle line in the high state prior to transmission/reception of a message;



NOTE: The Serial Communications Data Register (SCDR) is controlled by the internal R/$\overline{W}$ signal. It is the transmit data register when written and received data register when read.

Figure 5. SCI Block Diagram

2) A start bit that is transmitted/received, indicating the start of each character;
3) Data that is transmitted and received least-significant bit (LSB) first;
4) A stop bit (tenth or eleventh bit set to logic one), which indicates the frame is complete; and
5) A break defined as the transmission or reception of a logic zero for some multiple of frames.

Selection of the word length is controlled by the M bit in serial communications control register 1 (SCCR1).

TRANSMIT OPERATION

The SCI transmitter includes a parallel data register and a serial shift register. This double-buffered system allows a character to be shifted out serially while another character is waiting in the transmit data register to be transferred into the serial shift register. The output of the serial shift register is applied to PD1 as long as transmission is in progress or the transmit enable bit is set.

RECEIVE OPERATION

Data is received in a serial shift register and is transferred to a parallel receive data register as a complete word. This double-buffered system allows a character to be shifted in serially while another character is already in the receive data register. An advanced data recovery scheme is used to distinguish valid data from noise in the serial data stream. The data input is selectively sampled to detect receive data, and a majority voting circuit determines the value and intergrity of each bit.

WAKE-UP FEATURE

The wake-up feature reduces SCI service overhead in multiple receiver systems. Software for each receiver evaluates the first character(s) of each message. If the message is intended for a different receiver, the SCI can be placed in a sleep mode, disabling the rest of the message from generating requests for service. Whenever a new message begins, logic causes the sleeping receivers to awaken and evaluate the initial character(s) of the new message. Two methods of wake up are available: idle-line wake up or address mark wake up. In idle-line wake up, a sleeping receiver wakes up as soon as the RxD line becomes idle. In the address mark wake up, a "one" in the most-significant bit (MSB) of a character is used to indicate that the message is an address that wakes up a sleeping receiver.

SCI REGISTERS

The following paragraphs describe the operations of the five registers used in the SCI.

Serial Communications Data Registers (SCDR)

The SCDR performs two functions: as the receive data register when it is read and as the transmit data register when it is written. Figure 5 shows the SCDR as two separate registers.

Serial Communications Control Register 1 (SCCR1)

The SCCR1 provides the control bits to determine word length and select the method used for the wake-up feature.

7	6	5	4	3	2	1	0
R8	T8	0	M	WAKE	0	0	0
RESET: U	U	0	0	0	0	0	0

R8 — Receive Data Bit 8
If the M bit is set, this bit provides a storage location for the ninth bit in the receive data character.
T8 — Transmit Data Bit 8
If the M bit is set, this bit provides a storage location for the ninth bit in the transmit data character.
Bit 5 — Not Implemented
This bit always reads zero.
M — SCI Character Length
1 = 1 start bit, 9 data bits, 1 stop bit
0 = 1 start bit, 8 data bits, 1 stop bit
WAKE — Wake-Up Method Select
1 = Address mark
0 = Idle line
Bits 2-0 — Not Implemented
These bits always read zero.

Serial Communications Control Register 2 (SCCR2)

The SCCR2 provides the control bits that enable/disable individual SCI functions.

7	6	5	4	3	2	1	0
TIE	TCIE	RIE	ILIE	TE	RE	RWU	SBK
RESET: 0	0	0	0	0	0	0	0

TIE — Transmit Interrupt Enable
1 = SCI interrupt if TDRE = 1
0 = TDR interrupts disabled
TCIE — Transmit-Complete Interrupt Enable
1 = SCI interrupt if TC = 1
0 = TC interrupts disabled
RIE — Receive Interrupt Enable
1 = SCI interrupt if RDRF or OR = 1
0 = RDRF or OR interrupt disabled
ILIE — Idle-Line Interrupt Enable
1 = SCI interrupt if IDLE = 1
0 = IDLE interrupts disabled
TE — Transmit Enable
1 = Transmit shift register output is applied to the TxD line
0 = PD1 pin reverts to general-purpose I/O as soon as current transmitter activity finishes.
RE — Receive Enable
1 = Receiver enabled
0 = Receiver disabled and RDRF, IDLE, OR, NF, and FE interrupts are inhibited
RWU — Receiver Wake Up
When set by user's software, this bit puts the receiver to sleep and enables the "wake-up" function. If the WAKE bit is zero, RWU is cleared by the SCI logic after receiving 10 (M = 0) or 11 (M = 1) consecutive ones. If WAKE is one, RWU is cleared by the SCI logic after receiving a data word whose MSB is set.
SBK — Send Break
If this bit is toggled set and cleared, the transmitter sends 10 (M = 0) or 11 (M = 1) zeros and then reverts to idle or to sending data. If SBK remains set, the transmitter will continually send whole frames of zeros (sets of 10 or 11) until cleared.

Serial Communications Status Register (SCSR)

The SCSR provides inputs to the interrupt logic circuits for generation of the SCI system interrupts.

7	6	5	4	3	2	1	0
TDRE	TC	RDRF	IDLE	OR	NF	FE	0
RESET: 1	1	0	0	0	0	0	0

TDRE — Transmit Data Register Empty
1 = Automatically set when contents of the serial communications data register was transferred to the transmit serial shift register
0 = Cleared by a read of SCSR (with TDRE = 1) followed by a write to SCDR
TC — Transmit Complete
1 = Automatically set when all data frame, preamble, or break condition transmissions are complete
0 = Cleared by a read of SCSR (with TC = 1) followed by a write to SCDR
RDRF — Receive Data Register Full
1 = Automatically set when a character is transferred from the receiver shift register to the SCDR
0 = Cleared by a read of SCSR (with RDRF = 1) followed by a read of SCDR
IDLE — Idle-Line Detect
This bit is inhibited while RWU = 1.
1 = Automatically set when the receiver serial input becomes idle after having been active
0 = Cleared by a read of SCSR (with IDLE = 1) followed by a read of SCDR
OR — Overrun Error
1 = Automatically set when a new character cannot transfer from the receive shift register because the character in SCDR has not been read
0 = Cleared by a read of SCSR (with OR = 1) followed by a read of SCDR
NF — Noise Flag
1 = Automatically set when majority voting logic does not bind unanimous agreement of all samples in any bit in the received frame
0 = Cleared by a read of SCSR (with NF = 1) followed by a write to SCDR
FE — Framing Error
1 = Automatically set when a logic 0 is detected where a stop bit was expected
0 = Cleared by a read of SCSR (with FE = 1) followed by a read of SCDR
Bit 0 — Not Implemented
This bit always reads zero.

Baud-Rate Register (BAUD)

This register is used to select different baud rates that may be used as the rate control for the receiver and transmitter.

7	6	5	4	3	2	1	0
TCLR	0	SCP1	SCP0	RCKB	SCR2	SCR1	SCR0
RESET: 0	0	0	0	0	U	U	U

TCLR — Clear Baud-Rate Counters (Test)
This bit is used to clear the baud-rate counter chain during factory testing. TCLR is zero and cannot be set while in normal operating modes.
Bit 6 — Not Implemented
This bit always reads zero.
SCP1 and SCP0 — SCI Baud-Rate Prescaler Selects
These bits control a prescaler whose output provides the input to a second divider which is controlled by the SCR2-SCR0 bits. Refer to Table 4.

RCKB — SCI Baud-Rate Clock Check (Test)
This bit is used during factory testing to enable the exclusive-OR of the receiver clock and transmitter clock to be driven out the TxD pin. RCKB is zero and cannot be set while in normal operating modes.
SCR2-SCR0 — SCI Baud-Rate Selects
These bits select the baud rate for both the transmitter and the receiver. The prescaler output selected by SCP1 and SCP0 is further divided by the setting of these bits. Refer to Table 5.

Table 4. Prescaler Highest Baud-Rate Frequency Output

SCP Bit 1	SCP Bit 0	Clock* Divided By	Crystal Frequency (MHz) 8.3886	8.0	4.9152	4.0	3.6864
0	0	1	131.072 K Baud	125.000 K Baud	76.80 K Baud	62.50 K Baud	57.60 K Baud
0	1	3	43.690 K Baud	41.666 K Baud	25.60 K Baud	20.833 K Baud	19.20 K Baud
1	0	4	32.768 K Baud	31.250 K Baud	19.20 K Baud	15.625 K Baud	14.40 K Baud
1	1	13	10.082 K Baud	9600 Baud	5.907 K Baud	4800 Baud	4430 Baud

*The clock in the "Clock Divide By" column is the internal processor clock.

MC68HC811E2

Table 5. Transmit Baud-Rate Output for a Given Prescaler Output

SCR Bit			Divided By	Representative Highest Prescaler Baud-Rate Output				
2	1	0		131.072 K Baud	32.768 K Baud	76.80 K Baud	19.20 K Baud	9600 Baud
0	0	0	1	131.072 K Baud	32.768 K Baud	76.80 K Baud	19.20 K Baud	9600 Baud
0	0	1	2	65.536 K Baud	16.384 K Baud	38.40 K Baud	9600 Baud	4800 Baud
0	1	0	4	32.768 K Baud	8.192 K Baud	19.20 K Baud	4800 Baud	2400 Baud
0	1	1	8	16.384 K Baud	4.096 K Baud	9600 Baud	2400 Baud	1200 Baud
1	0	0	16	8.192 K Baud	2.048 K Baud	4800 Baud	1200 Baud	600 Baud
1	0	1	32	4.096 K Baud	1.024 K Baud	2400 Baud	600 Baud	300 Baud
1	1	0	64	2.048 K Baud	512 Baud	1200 Baud	300 Baud	150 Baud
1	1	1	128	1.024 K Baud	256 Baud	600 Baud	150 Baud	75 Baud

3

SERIAL PERIPHERAL INTERFACE

The serial peripheral interface (SPI) is a high-speed synchronous serial I/O system. The transfer rate is software selectable up to one-half of the MCU E clock rate. The SPI may be used for simple I/O expansion or to allow several MCUs to be interconnected in a multimaster configuration. Clock phase and polarity are software programmable to allow direct compatibility with a large number of peripheral devices.

Four basic signal lines are associated with the SPI system. These are the master-out-slave-in (MOSI), the master-in-slave-out (MISO), the serial clock (SCK), and the slave select ($\overline{SS}$). When data is written to the SPI data register of a master device, a transfer is automatically initiated. A series of eight SCK clock cycles are generated to synchronize data transfer.

When a master device transmits data to a slave device via the MOSI line, the slave device responds by sending data to the master device via the MISO line. This implies full duplex transmission with both data out and data in synchronized with the same clock signal. The byte transmitted is replaced by the byte received, thereby eliminating the need for separate transmit-empty and receiver-full status bits. Figure 6 shows a block diagram of the SPI.

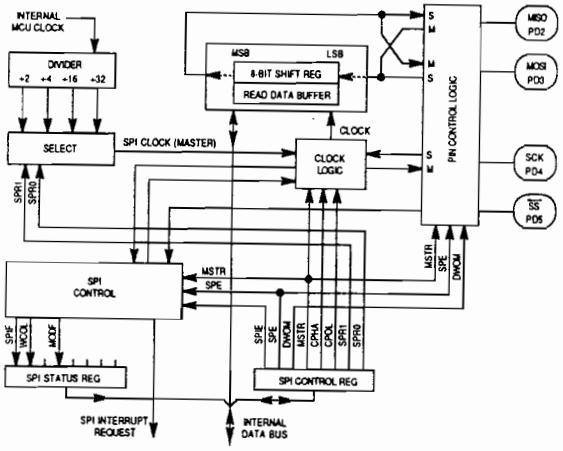


Figure 6. SPI Block Diagram

SPI REGISTERS

There are three registers in the SPI that provide control, status, and data-storage functions. These registers are described in the following paragraphs.

Serial Peripheral Control Register (SPCR) $1028

7	6	5	4	3	2	1	0
SPIE	SPE	DWOM	MSTR	CPOL	CPHA	SPR1	SPR0
RESET: 0	0	0	0	0	1	U	U

SPIE — Serial Peripheral Interrupt Enable
1 = SPI interrupt if SPIF = 1
0 = SPIF interrupts disabled

SPE — Serial Peripheral System Enable
1 = SPI system on
0 = SPI system off

DWOM — Port D Wire-OR Mode Option
This bit affects all six port D pins together.
1 = Port D outputs act as open-drain outputs
0 = Port D outputs are normal CMOS outputs

MSTR — Master Mode Select
1 = Master mode
0 = Slave mode

CPOL — Clock Polarity
This bit selects the polarity of the SCK clock.
1 = SCK line idles high
0 = SCK line idles low

CPHA — Clock Phase
This bit selects one of two fundamentally different clock protocols. Refer to Figure 7.
If CPHA = 0, transfer begins when $\overline{SS}$ goes low and ends when $\overline{SS}$ goes high after eight clock cycles on SCK. If CPHA = 1, transfer begins the first time SCK becomes active while $\overline{SS}$ is low and ends when the SPIF flag gets set.

SPR1 and SPR0 — SPI Clock Rate Select
These two bits select one of four baud rates to be used as SCK if the SPI is set as the master. They have no effect in the slave mode.

SPR1	SPR0	Internal Processor Clock Divide By
0	0	2
0	1	4
1	0	16
1	1	32

Serial Peripheral Status Register (SPSR) $1029

7	6	5	4	3	2	1	0
SPIF	WCOL	0	MODF	0	0	0	0
RESET: 0	0	0	0	0	0	0	0

SPIF — SPI Transfer Complete Flag
1 = Automatically set when data transfer is complete between processor and external device
0 = Cleared by a read of SPSR (with SPIF = 1), followed by an access (read or write) of the SPDR

WCOL — Write Collision
1 = Automatically set when an attempt is made to write to the SPI data register while data is being transferred

3

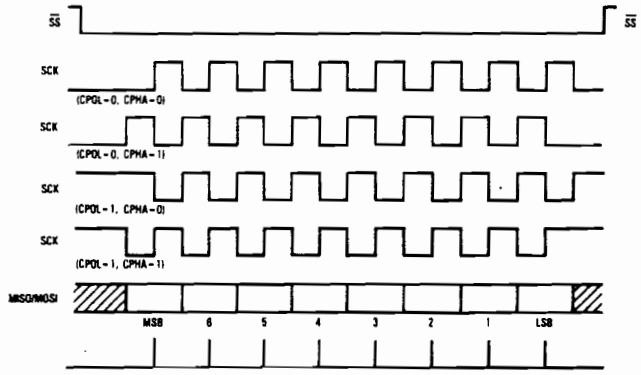


Figure 7. Data Clock Timing Diagram

0 = Cleared by a read of SPSR (with WCOL = 1), followed by an access (read or write) of the SPDR

Bit 5 — Not Implemented

This bit always reads zero.

MODF — Mode Fault

This bit indicates the possibility of a multi-master conflict for system control and therefore allows a proper exit from system operation to a reset or default system state.

1 = Automatically set when a master device has its $\overline{SS}$ pin pulled low

0 = Cleared by a read of SPSR (with MODF = 1), followed by a write to the SPCR.

Bits 3-0 — Not Implemented

These bits always read zero.

Serial Peripheral Data I/O Register (SPDR)

This register is used to transmit and receive data on the serial bus. A write to this register in a master will initiate transmission/reception of another byte. A slave writes data to this register for later transmission to a master. When transmission is complete, the SPIF status bit is set in both the master and slave device. When a read is performed on the SPDR, a buffer is actually being read. The first SPIF must be cleared by the time a second transfer of data from the shift register to the read buffer is initiated, or an overrun condition will exist. In case of an overrun, the byte causing the overrun is lost.

ANALOG-TO-DIGITAL CONVERTER

The MCU contains an 8-channel, multiplexed-input, successive approximation, analog-to-digital (A/D) converter with sample and hold. Two dedicated lines (V_{RL}, and V_{RH}) are provided for the reference supply voltage input. These pins are used instead of the device power pins to increase the accuracy of the A/D conversion.

The 8-bit A/D conversions of the MCU are accurate to within ±1 LSB (±1/2 LSB quantizing errors and ±1/2 LSB all other errors combined).Each conversion is accomplished in 32 MCU E-clock cycles. An internal control bit allows selection of an internal conversion clock oscillator that allows the A/D to be used with very low MCU clock rates. A typical conversion cycle requires 16 microseconds to complete at a 2-MHz bus frequency.

Four result registers are included to further enhance the A/D subsystem along with control logic to control conversion activity automatically. A single write instruction selects one of four conversion sequences, resulting in a conversion complete flag after the first four conversions. The sequences are as follows:

1) Convert one channel four times and stop, sequential results placed in the result registers.
2) Convert one group of four channels and stop, each result register is dedicated to one channel.
3) Convert one channel continuously, updating the result registers in a round-robin fashion.
4) Convert one group of four channels (round-robin fashion) continuously, each result register is dedicated to one channel.

INSTRUCTION SET

The MCU can execute all of the M6800 and M6801 instructions. In addition to these instructions, 91 new opcodes are provided by the paged opcode map. These instructions can be divided into five different types: 1) accumulator and memory, 2) index register and stack pointer, 3) jump, branch, and program control, 4) bit manipulation, and 5) condition code register instructions. The following paragraphs briefly explain each type.

ACCUMULATOR/MEMORY INSTRUCTIONS

Most of these instructions use two operands. One operand is either the accumulator or the index register. The other operand is obtained from memory using one of the addressing modes. The accumulator/memory instructions can be divided into four subgroups: 1) load/store/transfer, 2) arithmetic/math, 3) logical, and 4) shift/rotate. The following paragraphs describe the different groups of accumulator/memory instructions.

Load/Store/Transfer

Refer to the following table for load/store/transfer instructions.

Function	Mnemonic
Clear Memory Byte	CLR
Clear Accumulator A	CLRA
Clear Accumulator B	CLRB
Load Accumulator A	LDAA
Load Accumulator B	LDAB
Load Double Accumulator D	LDD
Push A onto Stack	PSHA
Push B onto Stack	PSHB
Pull A from Stack	PULA
Pull B from Stack	PULB
Store Accumulator A	STAA
Store Accumulator B	STAB
Store Accumulator D	STD
Transfer A to B	TAB
Transfer A to CC Register	TAP
Transfer B to A	TBA
Transfer CC Register to A	TPA
Exchange D with X	XGDX
Exchange D with Y	XGDY

Logical

This group is used to make comparisions, decisions, and extractions of data. Refer to the following list for the logical instructions.

Function	Mnemonic
AND A with Memory	ANDA
AND B with Memory	ANDB

— Continued—

BC780

Function	Mnemonic
Bit(s) Test A with Memory	BITA
Bit(s) Test B with Memory	BITB
1's Complement Memory Byte	COM
1's Complement A	COMA
1's Complement B	COMB
Exclusive OR A with Memory	EORA
Exclusive OR B with Memory	EORB
OR Accumulator A (Inclusive)	ORAA
OR Accumulator B (Inclusive)	ORAB

Shift/Rotate

The shift and rotate instructions automatically operate through the carry bit, which allows easy extension to multiple bytes. Refer to the following list for the shift/rotate instructions.

Function	Mnemonic
Arithmetic Shift Left	ASL
(Logical Shift Left)	(LSL)
Arithmetic Shift Left A	ASLA
(Logical Shift Left Accumulator A)	(LSLA)
Arithmetic Shift Left B	ASLB
(Logical Shift Left Accumulator B)	(LSLB)
Arithmetic Shift Left Double	ASLD
(Logical Shift Left Double)	(LSLD)
Arithmetic Shift Right	ASR
Arithmetic Shift Right A	ASRA
Arithmetic Shift Right B	ASRB
Logical Shift Right	LSR
Logical Shift Right Accumulator A	LSRA
Logical Shift Right Accumulator B	LSRB
Logical Shift Right Double	LSRD
Rotate Left	ROL
Rotate Left Accumulator A	ROLA
Rotate Left Accumulator B	ROLB
Rotate Right	ROR
Rotate Right Accumulator A	RORA
Rotate Right Accumulator B	RORB

Arithmetic/Math

Refer to the following table for the arithmetic/math instructions.

Function	Mnemonic
Add Accumulators	ABA
Add B to X	ABX
Add B to Y	ABY
Add with Carry to A	ADCA
Add with Carry to B	ADCB
Add Memory to A	ADDA
Add Memory to B	ADDB
Add 16-Bit to D	ADDD
Compare A to B	CBA
Compare A to Memory	CMPA
Compare B to Memory	CMPB
Compare D to Memory (16 Bit)	CPD
Decimal Adjust A	DAA
Decrement Memory Byte	DEC
Decrement Accumulator A	DECA
Decrement Accumulator B	DECB
Fractional Divide 16 × 16	FDIV
Integer Divide 16 × 16	IDIV
Increment Memory Byte	INC
Increment Accumulator A	INCA
Increment Accumulator B	INCB
Multiply 8 × 8	MUL
2's Complement Memory Byte	NEG
2's Complement A	NEGA
2's Complement B	NEGB
Subtract B from A	SBA
Subtract with Carry from A	SBCA
Subtract with Carry from B	SBCB
Subtract Memory from A	SUBA
Subtract Memory from B	SUBB
Subtract Memory from D	SUBD
Test for Zero or Minus	TST
Test for Zero or Minus A	TSTA
Test for Zero or Minus B	TSTB

INDEX-REGISTER AND STACK-POINTER INSTRUCTIONS

These instructions provide a method for storing data and for manipulation of index register, stack pointer, and individual segments of data within the register and stack pointer. Refer to the following list for the index-register and stack-pointer instructions.

Function	Mnemonic
Add B to X	ABX
Add B to Y	ABY
Compare X to Memory (16 Bit)	CPX
Compare Y to Memory (16 Bit)	CPY

— Continued —

Function	Mnemonic
Decrement Stack Pointer	DES
Decrement Index Register X	DEX
Decrement Index Register Y	DEY
Increment Stack Pointer	INS
Increment Index Register X	INX
Increment Index Register Y	INY
Load Index Register X	LDX
Load Index Register Y	LDY
Load Stack Pointer	LDS
Push X onto Stack (Low First)	PSHX
Push Y onto Stack (Low First)	PSHY
Pull X from Stack (High First)	PULX
Pull Y from Stack (High First)	PULY
Store Stack Pointer	STS
Store Index Register X	STX
Store Index Register Y	STY
Transfer Stack Pointer to X	TSX
Transfer Stack Pointer to Y	TSY
Transfer X to Stack Pointer	TXS
Transfer Y to Stack Pointer	TYS
Exchange D with X	XGDX
Exchange D with Y	XGDY

BIT-MANIPULATION INSTRUCTIONS

The MCU is capable of setting or clearing any bit residing in the first 256 bytes of the memory space in direct address mode. The MCU can use any bit in the 64K memory map, and all bit-manipulation instructions can be used with direct or index (x or y) addressing modes. Software can configure the memory map so that internal RAM, and/or internal registers, or external memory space can occupy these addresses. The bit-manipulation instructions use an 8-bit mask, which allows simultaneous operations on any combination of bits in a location. Refer to the following list for the bit-manipulation instructions.

Function	Mnemonic
Clear Bit(s)	BCRL
Branch if Bit(s) Clear	BRCRL
Branch if Bit(s) Set	BRSET
Set Bit(s)	BSET

JUMPS/BRANCHES/PROGRAM-CONTROL INSTRUCTIONS

These instructions provide techniques for modifying the normal sequence of the program for conditional and unconditional branching. Refer to the following list for the jump/branch/program-control instructions.

Function	Mnemonic
Branch if Carry Clear	BCC
(Branch if Higher or Same)	(BHS)
Branch if Carry Set	BCS
(Branch if Lower)	(BLO)
Branch if = zero	BEQ
Branch if ≥ zero	BGE
Branch if > zero	BGT
Branch if Higher	BHI
Branch if ≤ Zero	BLE
Branch if Lower or Same	BLS
Branch if < Zero	BLT
Branch if Minus	BMI
Branch if not = Zero	BNE
Branch if Plus	BPL
Branch Always	BRA
Branch if Bit(s) Clear	BRCLR
Branch Never	BRN
Branch if Bit(s) Set	BRSET
Branch to Subroutine	BSR
Branch if Overflow Clear	BVC
Branch if Overflow Set	BVS
Jump	JMP
Jump to Subroutine	JSR
No Operation	NOP
Return from Interrupt	RTI
Return from Subroutine	RTS
Stop Internal Clocks	STOP
Software Interrupt	SWI
Test Operation (Test Mode Only)	TEST
Wait for Interrupt	WAI

CONDITION-CODE-REGISTER INSTRUCTIONS

These instructions are register reference instructions and are used to control processor operation during program execution. Refer to the following list for the condition-code-register instructions.

Function	Mnemonic
Clear Carry Bit	CLC
Clear Interrupt Mask	CLI
Clear Overflow Flag	CLV
Set Carry	SEC
Set Interrupt Mask	SEI
Set Overflow Flag	SEV
Transfer A to CC Register	TAP
Transfer CC Register to A	TPA

OPCODE MAP SUMMARY

Table 6 is an opcode map for the instructions used on the MCU.

ADDRESSING MODES

The MCU uses six different addressing modes to provide the programmer with an opportunity to optimize the code for all situations. Some instructions require an additional byte before the opcode to accommodate a multipage opcode map; this byte is called a prebyte.

The term "effective address" (EA) is used in describing the various addressing modes. Effective address is defined as the address from which the argument for an instruction is fetched or stored. The following paragraphs describe the different addressing modes.

IMMEDIATE

In the immediate addressing mode, the operand is contained in the byte immediately following the opcode. These are two, three, or four (if prebyte is required) byte instructions.

DIRECT

In the direct addressing mode, the least-significant byte of the operand address is contained in a single byte following the opcode and the most-significant byte of an address is assumed to be $00. Direct addressing allows the user to directly address $0000 through $00FF using two-byte instructions, and execution time is reduced by eliminating the additional memory access. In most applications, this 256-byte area is reserved for frequently referenced data. In the MCU, software can configure the memory map so that internal RAM, and/or internal registers, or external memory space can occupy these addresses.

EXTENDED

In the extended addressing mode, the effective address of the argument is contained in the two bytes following the opcode byte. These are three or four (if prebyte is required) byte instructions: one or two for the opcode and two for the effective address.

INDEXED

In the indexed addressing mode, one of the index registers (X or Y) is used in calculating the effective address. In this case, the effective address is variable and depends on two factors: 1) the current contents of the index register (X or Y) being used, and 2) the 8-bit unsigned offset contained in the instruction. This addressing mode allows referencing any memory location in the 64K byte address space. These are usually two or three (if prebyte is required) byte instructions, the opcode plus the 8-bit offset.

RELATIVE

The relative addressing mode is only used in branch instructions. In relative addressing, the contents of the 8-bit signed byte (the offset) following the opcode is added to the PC if, and only if, the branch conditions are true. Otherwise, control proceeds to the next instruction. These are usually two-byte instructions.

INHERENT

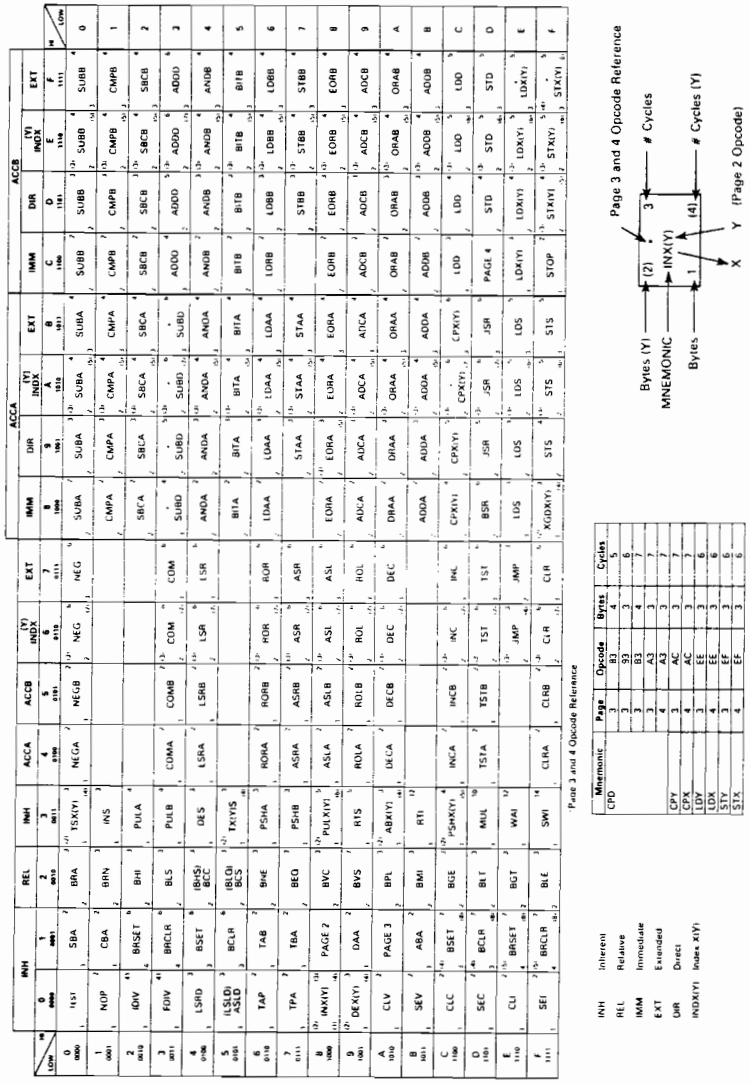
In the inherent addressing mode, all the information necessary to execute the instruction is contained in the opcode. Operations specifying only the index register or accumulator as well as the control instruction with no other arguments are included in this mode. These instructions are one- or two-byte instructions.

PREBYTE

To expand the number of instructions used in the MCU, a prebyte instruction has been added to certain instructions. The instructions affected are usually associated with index register Y. Accessing opcodes from page 2, 3, or 4 would require a prebyte instruction.

3

Table 6. Opcode Map



3

ELECTRICAL SPECIFICATIONS

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{DD}	−0.3 to +7.0	V
Input Voltage	V_{in}	−0.3 to +7.0	V
Operating Temperature Range MC68HC811E2 MC68HC811E2V MC68HC811E2M	T_A	T_L to T_H −40 to 85 −40 to 105 −40 to 125	°C
Storage Temperature Range	T_{stg}	−55 to 150	°C
Current Drain per Pin* Excluding V_{DD}, V_{SS}, V_{RH}, and V_{RL}	I_D	25	mA

*One pin at a time, observing maximum power dissipation limits.

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (e.g., either GND or V_{DD}).

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Unit
Thermal Resistance Plastic 52-Pin Quad Pack (PLCC)	θ_{JA}	50	°C/W

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \theta_{JA}) \quad (1)$$

where:

T_A = Ambient Temperature, °C
θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W
P_D = $P_{INT} + P_{I/O}$
P_{INT} = $I_{DD} \times V_{DD}$, Watts — Chip Internal Power
$P_{I/O}$ = Power Dissipation on Input and Output Pins, Watts — User Determined

For most applications $P_{I/O} < P_{INT}$ and can be neglected.

The following is an approximate relationship between P_D and T_J (if $P_{I/O}$ is neglected):

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations (1) and (2) for K gives:

$$K = P_D \cdot (T_A + 273°C) + \theta_{JA} \cdot P_D^2 \quad (3)$$

where K is a constant pertaining to the particular part. K can be determined from equation (3) by measuring P_D (at equilibrium) for a known T_A. Using this value of K, the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

DC ELECTRICAL CHARACTERISTICS (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, unless otherwise noted)

Characteristic		Symbol	Min	Max	Unit
Output Voltage I_{Load} = ± 10.0 µA (see Note 1)	All Outputs All Outputs Except $\overline{RESET}$ and MODA	V_{OL} V_{OH}	— V_{DD} – 0.1	0.1 —	V
Output High Voltage I_{Load} = – 0.8 mA, V_{DD} = 4.5 V (see Note 1)	All Outputs Except $\overline{RESET}$, XTAL, and MODA	V_{OH}	V_{DD} – 0.8	—	V
Output Low Voltage I_{Load} = 1.6 mA	All Outputs Except XTAL	V_{OL}	—	0.4	V
Input High Voltage	All Inputs Except $\overline{RESET}$ $\overline{RESET}$	V_{IH}	0.7 × V_{DD} 0.8 × V_{DD}	V_{DD} V_{DD}	V
Input Low Voltage	All Inputs	V_{IL}	V_{SS}	0.2 × V_{DD}	V
I/O Ports, Three-State Leakage $V_{in} = V_{IH}$ or V_{IL}	PA3, PA7, PC0-PC7, PD0-PD5, AS/STRA, MODA/$\overline{LIR}$, $\overline{RESET}$	I_{OZ}	—	± 10	µA
Input Current (see Note 2) $V_{in} = V_{DD}$ or V_{SS} $V_{in} = V_{DD}$ or V_{SS}	PA0-PA3, $\overline{IRQ}$, $\overline{XIRQ}$ MODB/V_{STBY}	I_{in}	— —	± 1 ± 10	µA
RAM Standby Voltage	Powerdown	V_{SB}	4.0	V_{DD}	V
RAM Standby Current	Powerdown	I_{SB}	—	20	µA
Total Supply Current (see Note 3) RUN: Single Chip Expanded Multiplexed WAIT: All Peripheral Functions Shut Down Single-Chip Mode Expanded Multiplexed Mode STOP: No Clocks, Single-Chip Mode		I_{DD} W_{IDD} S_{IDD}	 — — — — —	 15 27 6 10 100	 mA mA mA mA µA
Input Capacitance	PA0-PA3, PE0-PE7, $\overline{IRQ}$, $\overline{XIRQ}$, EXTAL PA7, PC0-PC7, PD0-PD5, AS/STRA, MODA/$\overline{LIR}$, $\overline{RESET}$	C_{in}	— —	8 12	pF
Power Dissipation	Single-Chip Mode Expanded-Multiplexed Mode	P_D	— —	85 150	mW

NOTES:
1. V_{OH} specification for $\overline{RESET}$ and MODA is not applicable because they are open-drain pins. V_{OH} specification not applicable to ports C and D in wire-OR mode.
2. See A/D specification for leakage current for port E.
3. All ports configured as inputs,
   $V_{IL} \leq 0.2$ V,
   $V_{IH} \geq V_{DD} - 0.2$ V,
   No dc loads,
   EXTAL is driven with a square wave, and
   t_{cyc} = 476.5 ns.

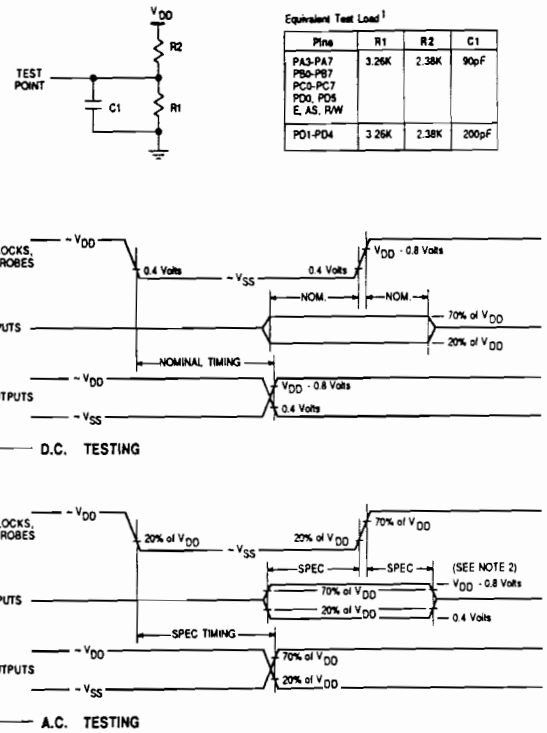


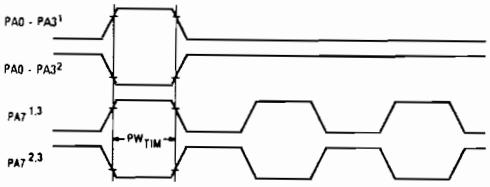
Figure 8. Test Methods

NOTES:
1. Full test loads are applied during all ac electrical test and ac timing measurements.
2. During ac timing measurements, inputs are driven to 0.4 volts and V_{DD} - 0.8 volts while timing measurements are taken at the 20% and 70% of V_{DD} points.

CONTROL TIMING (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

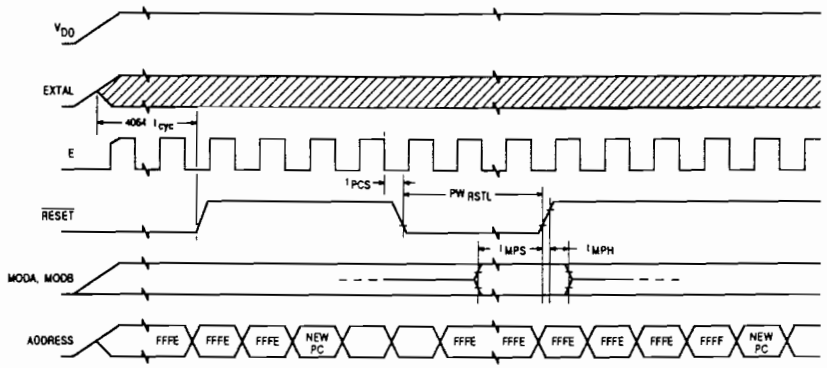
Characteristic	Symbol	1.0 MHz		2.0 MHz		2.1 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Frequency of Operation	f_o	dc	1.0	dc	2.0	dc	2.1	MHz
E Clock Period	t_{cyc}	1000	—	500	—	476	—	ns
Crystal Frequency	f_{XTAL}	—	4.0	—	8.0	—	8.4	MHz
External Oscillator Frequency	4 f_o	dc	4.0	dc	8.0	dc	8.4	MHz
Processor Control Setup Time (See Figures 10, 12, and 13) $t_{PCS} = 1/4\ t_{cyc} - 50$ ns	t_{PCS}	200	—	75	—	69	—	ns
Reset Input Pulse Width (see Note 1) and Figure 10) (To Guarantee External Reset Vector)	PW_{RSTL}	8	—	8	—	8	—	t_{cyc}
(Minimum Input Time; May be Preempted by Internal Reset)		1	—	1	—	1	—	
Mode Programming Setup Time (See Figure 10)	t_{MPS}	2	—	2	—	2	—	t_{cyc}
Mode Programming Hold Time (See Figure 10)	t_{MPH}	0	—	0	—	0	—	ns
Interrupt Pulse Width, $\overline{IRQ}$ Edge Sensitive Mode (See Figure 11 and 13) $PW_{IRQ} = t_{cyc} + 20$ ns	PW_{IRQ}	1020	—	520	—	496	—	ns
Wait Recovery Startup Time (See Figure 12)	t_{WRS}	—	4	—	4	—	4	t_{cyc}
Timer Pulse Width Input Capture, Pulse Accumulator Input (See Figure 9) $PW_{TIM} = t_{cyc} + 20$ ns	PW_{TIM}	1020	—	520	—	496	—	ns

NOTES:
1. $\overline{RESET}$ will be recognized during the first clock cycle it is held low. Internal circuitry then drives the pin low for four clock cycles, releases the pin, and samples the pin level two cycles later to determine the source of the interrupt. See **RESETS, INTERRUPT, AND LOW-POWER MODES** for details.
2. All timing is shown with respect to 20% V_{DD} and 70% V_{DD} unless otherwise noted.



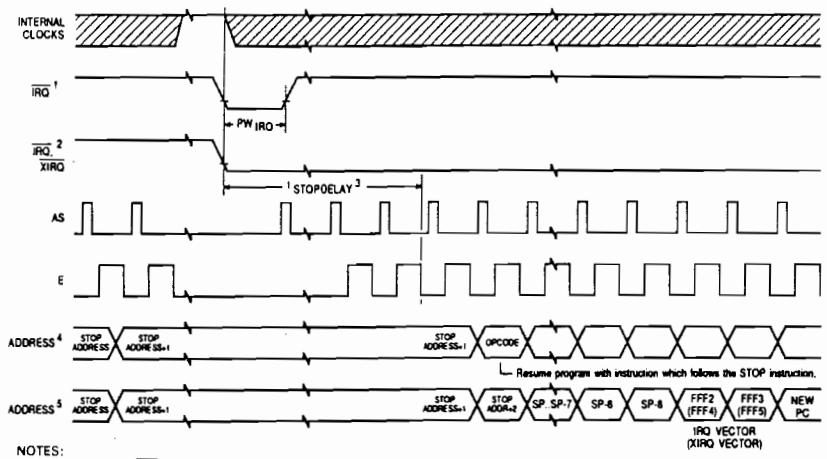
NOTES:
1. Rising edge sensitive input.
2. Falling edge sensitive input.
3. Maximum pulse accumulator clocking rate is E frequency divided by 2.

Figure 9. Timer Inputs Timing Diagram



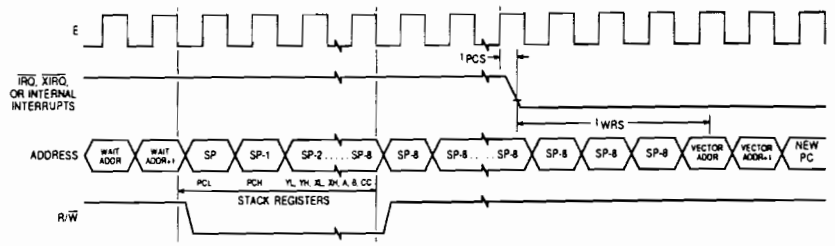
NOTE: Refer to Table 9-7 for pin states during $\overline{RESET}$

Figure 10. POR External Reset Timing Diagram



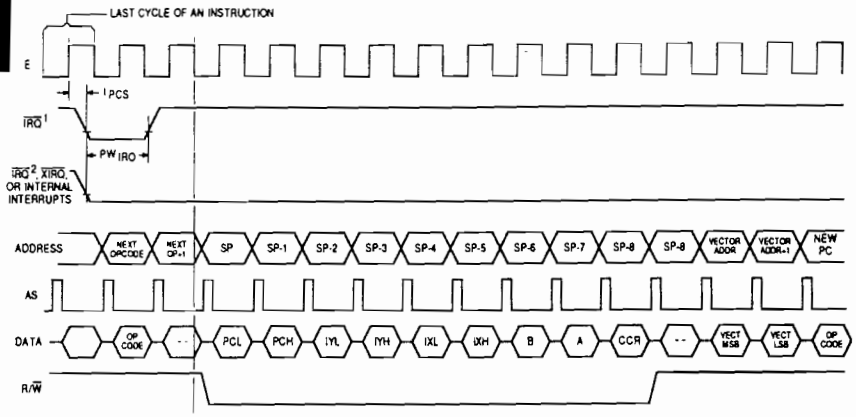
NOTES:
1. Edge sensitive $\overline{IRQ}$ pin (IRQE bit = 1)
2. Level sensitive $\overline{IRQ}$ pin (IRQE bit = 0)
3. $t_{STOPDELAY} = 4064\ t_{cyc}$ if DLY bit = 1 or 4 t_{cyc} if DLY = 0.
4. $\overline{XIQ}$ with X bit in CCR = 1.
5. $\overline{IRQ}$ or ($\overline{XIRQ}$ with X bit in CCR = 0.

Figure 11. Stop Recovery Timing Diagram



NOTES:
1. Refer to Table 9-7 for pin states during WAIT.
2. $\overline{\text{RESET}}$ will also cause recovery from WAIT.

Figure 12. WAIT Recovery from Interrupt Timing Diagram



NOTES:
1. Edge sensitive $\overline{\text{IRQ}}$ pin (IROE bit = 1).
2. Level sensitive $\overline{\text{IRQ}}$ pin (IROE bit = 0).

Figure 13. Interrupt Timing Diagram

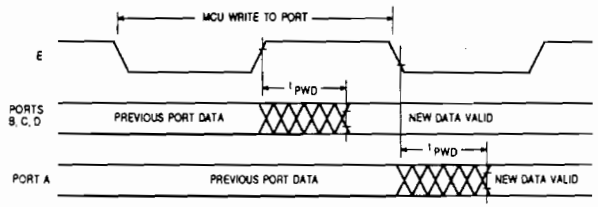
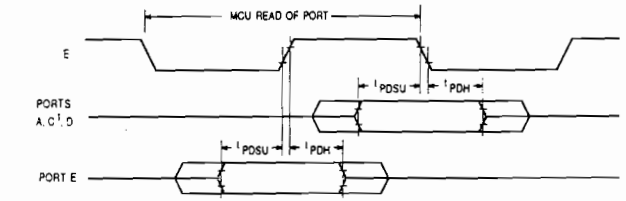


Figure 14. Port Write Timing Diagram



NOTE1. For non-latched operation of Port C.

Figure 15. Port Read Timing Diagram

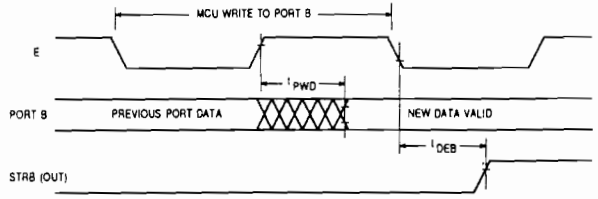


Figure 16. Simple Output Strobe Timing Diagram

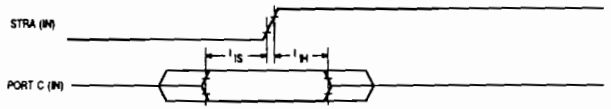
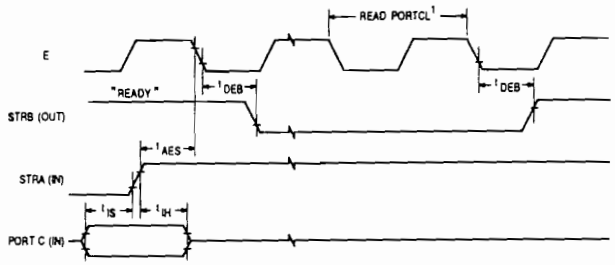
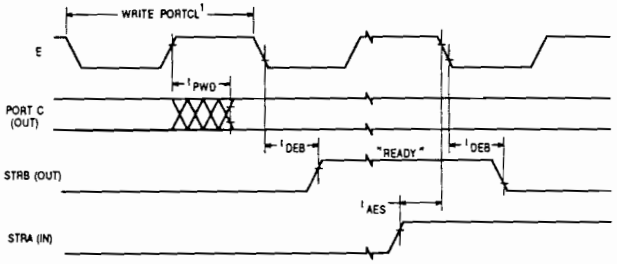


Figure 17. Simple Input Strobe Timing Diagram



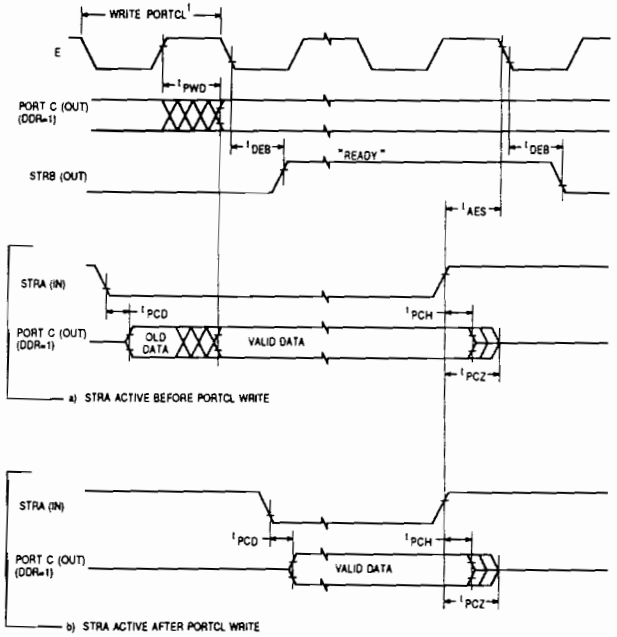
NOTES:
1. After reading PIOC with STAF set.
2. Figure shows rising edge STRA (EGA = 1) and high true STRB (INVB = 1).

Figure 18. Port C Input Handshake Timing Diagram



NOTES:
1. After reading PIOC with STAF set.
2. Figure shows rising edge STRA (EGA = 1) and high true STRB (INVB = 1).

Figure 19. Port C Output Handshake Timing Diagram



NOTES:
1. After reading PIOC with STAF set.
2. Figure shows rising edge STRA (EGA = 1) and high true STRB (INVB = 1).

Figure 20. Three-State Variation of Output Handshake Timing Diagram (STRA Enables Output Buffer)

PERIPHERAL PORT TIMING (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

Characteristic	Symbol	1.0 MHz		2.0 MHz		2.1 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Frequency of Operation (E Clock Frequency)	f_o	1.0	1.0	2.0	2.0	2.1	2.1	MHz
E Clock Period	t_{cyc}	1000	—	500	—	476	—	ns
Peripheral Data Setup Time (MCU Read of Ports A, C, D, and E) (See Figure 15)	t_{PDSU}	100	—	100	—	100	—	ns
Peripheral Data Hold Time (MCU Read of Ports A, C, D, and E) (See Figure 15)	t_{PDH}	50	—	50	—	50	—	ns
Delay Time, Peripheral Data Write (See Figures 14, 16, 18, and 19)	t_{PWD}							ns
MCU Write to Port A		—	150	—	150	—	150	
MCU Writes to Ports B, C, and D $t_{PWD} = 1.4\ t_{cyc} + 90$ ns		—	340	—	215	—	209	
Input Data Setup Time (Port C) (See Figures 17 and 18)	t_{IS}	60	—	60	—	60	—	ns
Input Data Hold Time (Port C) (See Figures 17 and 18)	t_{IH}	100	—	100	—	100	—	ns
Delay Time, E Fall to STRB $t_{DEB} = 1.4\ t_{cyc} + 100$ ns (See Figure 16, 18, 19, and 20)	t_{DEB}	—	350	—	225	—	219	ns
Setup Time, STRA Asserted to E Fall (see Note 1) (See Figures 18, 19, 20)	t_{AES}	0	—	0	—	0	—	ns
Delay Time, STRA Asserted to Port C Data Output Valid (See Figure 20)	t_{PCD}	—	100	—	100	—	100	ns
Hold Time, STRA Negated to Port C Data (See Figure 20)	t_{PCH}	10	—	10	—	10	—	ns
Three-State Hold Time (See Figure 20)	t_{PCZ}	—	150	—	150	—	150	ns

NOTES:
1. If this setup time is met, STRB will acknowledge in the next cycle. If it is not met, the response may be delayed one more cycle.
2. Port C and D timing is valid for active drive (CWOM and DWOM bits not set in PIOC and SPCR registers respectively).
3. All timing is shown with respect to 20% V_{DD} and 70% V_{DD} unless otherwise noted.

A/D CONVERTER CHARACTERISTICS (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, 750 kHz ≤ E ≤ 2.1 MHz, unless otherwise noted)

Characteristic	Parameter	Min	Absolute	Max	Unit
Resolution	Number of Bits Resolved by the A/D	8	—	—	Bits
Non-Linearity	Maximum Deviation from the Ideal A/D Transfer Characteristics	—	—	± 1/2	LSB
Zero Error	Difference Between the Output of an Ideal and an Actual A/D for Zero Input Voltage	—	—	± 1/2	LSB
Full-Scale Error	Difference Between the Output of an Ideal and an Actual A/D for Full-Scale Input Voltage	—	—	± 1/2	LSB
Total Unadjusted Error	Maximum Sum of Non-Linearity, Zero Error, and Full-Scale Error	—	—	± 1 2	LSB
Quantization Error	Uncertainty Due to Converter Resolution	—	—	± 1 2	LSB
Absolute Accuracy	Difference Between the Actual Input Voltage and the Full-Scale Weighted Equivalent of the Binary Output Code, All Error Sources Included	—	—	± 1	LSB
Conversion Range	Analog Input Voltage Range	V_{RL}	—	V_{RH}	V
V_{RH}	Maximum Analog Reference Voltage (see Note 2)	V_{RL}	—	V_{DD} + 0.1	V
V_{RL}	Minimum Analog Reference Voltage (see Note 2)	V_{SS} − 0.1	—	V_{RH}	V
ΔV_R	Minimum Difference between V_{RH} and V_{RL} (see Note 2)	3	—	—	V
Conversion Time	Total Time to Perform a Single Analog-to-Digital Conversion:				
	a. E Clock	—	32	—	t_{cyc}
	b. Internal RC Oscillator	—	—	t_{cyc} − 32	µs
Monotonicity	Conversion Result Never Decreases with an Increase in Input Voltage and has no Missing Codes		Guaranteed		
Zero-Input Reading	Conversion Result when $V_{in} = V_{RL}$	00	—	—	Hex
Full-Scale Reading	Conversion Result when $V_{in} = V_{RH}$	—	—	FF	Hex
Sample Acquisition Time	Analog Input Acquisition Sampling Time:				
	a. E Clock	—	12	—	t_{cyc}
	b. Internal RC Oscillator	—	—	12	µs
Sample/Hold Capacitance	Input Capacitance during Sample PE0-PE7	—	20 (Typ)	—	pF
Input Leakage	Input Leakage on A/D Pins PE0-PE7	—	—	400	nA
	V_{RL}, V_{RH}	—	—	1.0	µA

NOTES:
1. Source impedances greater than 10 KΩ will adversely affect accuracy, due mainly to input leakage.
2. Performance verified down to 2.5 V ΔV_R, but accuracy is tested and guaranteed at ΔV_R = 5 V ± 10%.

EXPANSION BUS TIMING (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, see Figure 21)

Num.	Characteristic	Symbol	1.0 MHz		2.0 MHz		2.1 MHz		Unit
			Min	Max	Min	Max	Min	Max	
	Frequency of Operation (E Clock Frequency)	f_o	1.0	1.0	2.0	2.0	2.1	2.1	MHz
1	Cycle Time	t_{cyc}	1000	—	500	—	476	—	ns
2	Pulse Width, E Low $PW_{EL} = 1/2\ t_{cyc} - 23$ ns	PW_{EL}	477	—	227	—	215	—	ns
3	Pulse Width, E High $PW_{EH} = 1/2\ t_{cyc} - 28$ ns	PW_{EH}	472	—	222	—	210	—	ns
4	E and AS Rise and Fall Time	t_r, t_f	—	20	—	20	—	20	ns
9	Address Hold Time $t_{AH} = 1/8\ t_{cyc} - 29.5$ ns see Note 1(a)	t_{AH}	95.5	—	33	—	30	—	ns
12	Non-Muxed Address Valid Time to E Rise $t_{AV} = PW_{EL} - (t_{ASD} + 80$ ns) see Note 1(b)	t_{AV}	281.5	—	94	—	85	—	ns
17	Read Data Setup Time	t_{DSR}	30	—	30	—	30	—	ns
18	Read Data Hold Time (Max = t_{MAD})	t_{DHR}	10	145.5	10	83	10	80	ns
19	Write Data Delay Time $t_{DDW} = 1/8\ t_{cyc} + 65.5$ ns see Note 1(a)	t_{DDW}	—	190.5	—	128	—	125	ns
21	Write Data Hold Time $t_{DHW} = 1/8\ t_{cyc} - 29.5$ ns see Note 1(a)	t_{DHW}	95.5	—	33	—	30	—	ns
22	Muxed Address Valid Time to E Rise $t_{AVM} = PW_{EL} - (t_{ASD} + 90$ ns) see Note 1(b)	t_{AVM}	271.5	—	84	—	75	—	ns
24	Muxed Address Valid Time to AS Fall $t_{ASL} = PW_{ASH} - 70$ ns	t_{ASL}	151	—	26	—	20	—	ns
25	Muxed Address Hold Time $t_{AHL} = 1/8\ t_{cyc} - 29.5$ ns see Note 1(b)	t_{AHL}	95.5	—	33	—	30	—	ns
26	Delay Time, E to AS Rise $t_{ASD} = 1/8\ t_{cyc} - 9.5$ ns see Note 1(a)	t_{ASD}	115.5	—	53	—	50	—	ns
27	Pulse Width, AS High $PW_{ASH} = 1/4\ t_{cyc} - 29$ ns	PW_{ASH}	221	—	96	—	90	—	ns
28	Delay Time, AS to E Rise $t_{ASED} = 1/8\ t_{cyc} - 9.5$ ns see Note 1(b)	t_{ASED}	115.5	—	53	—	50	—	ns
29	MPU Address Access Time see note 1(b) $t_{ACCA} = t_{AVM} + t_r + PW_{EH} - t_{DSR}$	t_{ACCA}	733.5	—	296	—	275	—	ns
35	MPU Access Time $t_{ACCE} = PW_{EH} - t_{DSR}$	t_{ACCE}	—	442	—	192	—	180	ns
36	Muxed Address Delay (Previous Cycle MPU Read) $t_{MAD} = t_{ASD} + 30$ ns see Note 1(a)	t_{MAD}	145.5	—	83	—	80	—	ns

NOTES:
1. Input clocks with duty cycles other than 50% will affect bus performance. Timing parameters affected by input clock duty cycle are identified by (a) and (b). To recalculate the approximate bus timing values, substitute the following expressions in place of 1.8 t_{cyc} in the above formulas where applicable:
   (a) (1-DC) × 1.4 t_{cyc}
   (b) DC × 1.4 t_{cyc}
   Where:
   DC is the decimal value of duty cycle percentage (high time)
2. All timing is shown with respect to 20% V_{DD} and 70% V_{DD} unless otherwise noted.

SERIAL PERIPHERAL INTERFACE (SPI) TIMING (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H, see Figure 22)

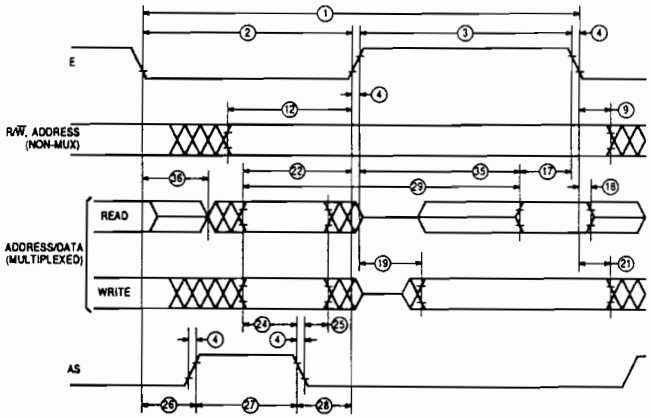
Num.	Characteristic	Symbol	Min	Max	Unit
	Operating Frequency Master Slave	 $f_{op(m)}$ $f_{op(s)}$	 dc dc	 0.5 2.1	f_{op} MHz
1	Cycle Time Master Slave	 $t_{cyc(m)}$ $t_{cyc(s)}$	 2.0 480	 — —	t_{cyc} ns
2	Enable Lead Time Master Slave	 $t_{lead(m)}$ $t_{lead(s)}$	 * 240	 — —	 ns ns
3	Enable Lag Time Master Slave	 $t_{lag(m)}$ $t_{lag(s)}$	 * 240	 — —	 ns ns
4	Clock (SCK) High Time Master Slave	 $t_{w(SCKH)m}$ $t_{w(SCKH)s}$	 340 190	 — —	 ns ns
5	Clock (SCK) Low Time Master Slave	 $t_{w(SCKL)m}$ $t_{w(SCKL)s}$	 340 190	 — —	 ns ns
6	Data Setup Time (Inputs) Master Slave	 $t_{su(m)}$ $t_{su(s)}$	 100 100	 — —	 ns ns
7	Data Hold Time (Inputs) Master Slave	 $t_{h(m)}$ $t_{h(s)}$	 100 100	 — —	 ns ns
8	Access Time (Time to Data Active from High-Impedance State) Slave	 t_a	 0	 120	 ns
9	Disable Time (Hold Time to High-Impedance State) Slave	 t_{dis}	 —	 240	 ns
10	Data Valid (After Enable Edge)**	$t_{v(s)}$	—	240	ns
11	Data Hold Time (Outputs) (After Enable Edge)	t_{ho}	0	—	ns
12	Rise Time (20% V_{DD} to 70% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, and SS)	 t_{rm} t_{rs}	 — —	 100 2.0	 ns µs
13	Fall Time (70% V_{DD} to 20% V_{DD}, C_L = 200 pF) SPI Outputs (SCK, MOSI, and MISO) SPI Inputs (SCK, MOSI, MISO, and SS)	 t_{fm} t_{fs}	 — —	 100 2.0	 ns µs

*Signal production depends on software.
**Assumes 200 pF load on all SPI pins.
NOTE:
1. All timing is shown with respect to 20% V_{DD} and 70% V_{DD}, unless otherwise noted.

EEPROM CHARACTERISTICS (V_{DD} = 5.0 Vdc ± 10%, V_{SS} = 0 Vdc, $T_A = T_L$ to T_H)

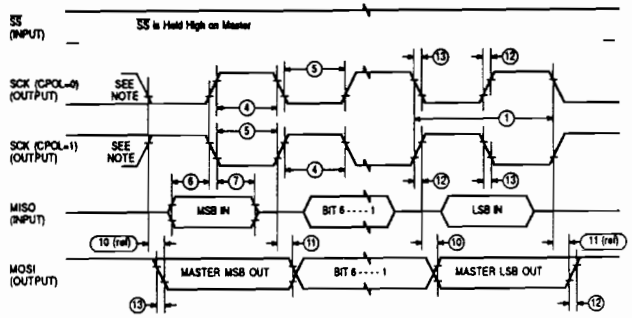
Characteristic		Temperature Range			Unit
		– 40 to 85°C	– 40 to 105°C	– 40 to 125°C	
Programming Time (see Note 1)	Under 1.0 MHz with RC Oscillator Enabled	10	15	20	ms
	1.0 to 2.0 MHz with RC Oscillator Disabled	20	Must Use RC	Must Use RC	
	2.0 MHz (or Anytime RC Oscillator Enabled)	10	15	20	
Erase Time (see Note 1)	Byte, Row, and Bulk	10	10	10	ms
Write/Erase Endurance (see Note 2)		10,000	10,000	10,000	Cycles
Data Retention (see Note 2)		10	10	10	Years

NOTES:
1. The RC oscillator must be enabled (by setting the CSEL bit in the OPTION register) for EEPROM programming and erasure when the E-clock frequency is below 1.0 MHz.
2. See current quarterly Reliability Monitor report for current failure rate information.



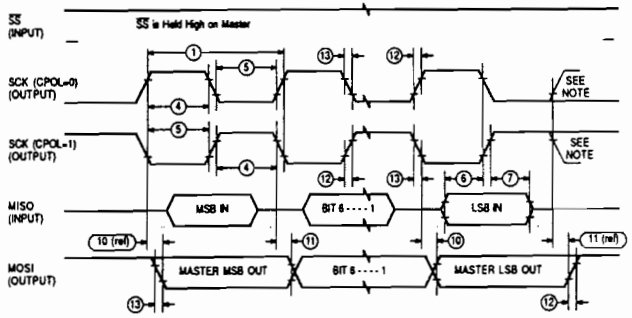
NOTE: Measurement points shown are 20% and 70% V_{DD}.

Figure 21. Expansion Bus Timing Diagram



NOTE: This first clock edge is generated internally but is not seen at the SCK pin.

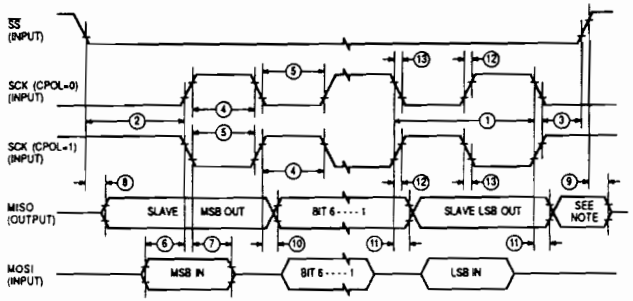
a) SPI MASTER TIMING (CPHA = 0)



NOTE: This last clock edge is generated internally but is not seen at the SCK pin.

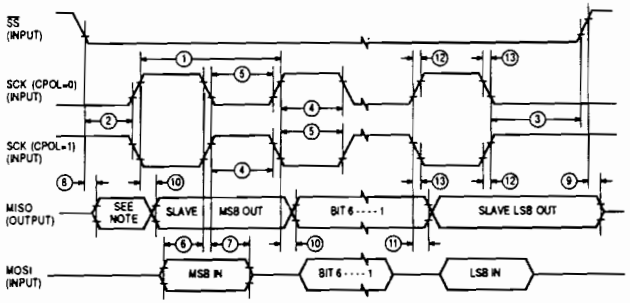
b) SPI MASTER TIMING (CPHA = 1)

Figure 22. SPI Timing Diagrams (Sheet 1 of 2)



NOTE: Not defined but normally MSB of character just received.

c) SPI SLAVE TIMING (CPHA = 0)



NOTE: Not defined but normally LSB of character previously transmitted.

d) SPI SLAVE TIMING (CPHA = 1)

Figure 22. SPI Timing Diagrams (Sheet 2 of 2)

ORDERING INFORMATION

The following information is required when ordering a custom MCU. The information may be transmitted to Motorola in the following media:

MS-DOS/PC-DOS disk file (360K)

EPROM(s): three 2532/2732 or two 2764

To initiate a ROM pattern for the MCU, it is necessary to first contact the local field-service office, a sales person, or a Motorola representative.

FLEXIBLE DISKS

Several types of flexible disks (MS-DOS™/PC-DOS disk file), programmed with the customer's program (positive logic sense for address and data), may be submitted for pattern generation. In either case, the diskette should be clearly labeled with the customer's name, date, project or product name, and the name of the file containing the pattern.

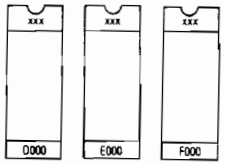
In addition to the program pattern, a file containing the program source code listing can be included. This data will be kept confidential and used to expedite the process in case of any difficulty with the pattern file.

MS-DOS is Microsoft's Disk Operating System. PC-DOS is IBM® Personal Computer Disk Operating System. Disk media submitted must be a standard density (360K), double-sided 5 1/4-inch compatible floppy diskette. The diskette must contain object file code in Motorola's S-record format. The S-record format is a character-based object file format generated by M68HC11 cross assemblers and linkers on IBM PC-style machines.

EPROMs

Three 2532/2732 or two 2764 type EPROM(s), programmed with the customer's program (positive logic sense for address and data), may be submitted for pattern generation. EPROMs must be clearly marked to indicate which EPROM corresponds to which address space. Figure 22 illustrates the markings for the three 2532/2732 EPROMs required to contain the customer's program.

All unused bytes, including the user's space, must be set to zero. For shipment to Motorola, EPROMs should be placed in a conductive IC carrier and packed securely. Styrofoam is not acceptable for shipment.



xxx = Customer ID

Figure 23. EPROM Marking

VERIFICATION MEDIA

All original pattern media (EPROMs or floppy disks) are filed for contractual purposes and are not returned. A computer listing of the ROM code will be generated and returned along with a listing verification form. The listing should be thoroughly checked, and the verification form should be completed, signed, and returned to Motorola. The signed verification form constitutes the contractual agreement for creation of the customer mask. To aid in the verification process, Motorola will program ***customer supplied*** blank EPROM(s) or DOS disks from the data file used to create the custom mask.

ROM VERIFICATION UNITS (RVUs)

Ten MCUs containing the customer's ROM pattern will be sent for program verification. These units will have been made using the custom mask, but are for the purpose of ROM verification only. For expediency, the MCUs are unmarked, packaged in ceramic, and tested with five volts at room temperature. These RVUs are free with the minimum-order quantity, but are not production parts. These RVUs are not guaranteed by Motorola Quality Assurance.

ORDERING INFORMATION

The following table provides ordering information pertaining to the package type, temperature, and MC part numbers for the MC68HC811E2 HCMOS single-chip microcontroller devices.

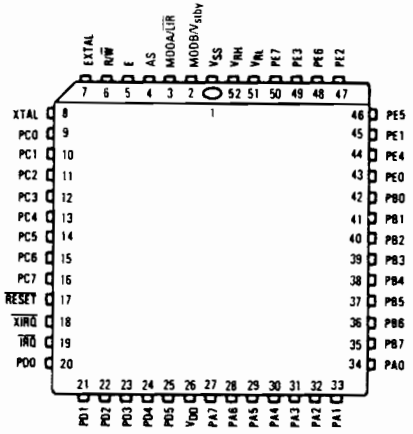
Package Type	Temperature	CONFIG	MC Part Number
PLCC (FN Suffix)	−40° to +85°C	$FF	MC68HC811E2FN
	−40 to +105°C	$FF	MC68HC811E2VFN
	−40 to +125°C	$FF	MC68HC811E2MFN

MS-™DOS is a trademark of Microsoft, Inc.

IBM is a registered trademark of International Business Machines Corporation.

PIN ASSIGNMENTS

52-Lead Quad Package



MC146818

Advance Information

REAL-TIME CLOCK PLUS RAM (RTC)

The MC146818 Real-Time Clock plus RAM is a peripheral device which includes the unique MOTEL concept for use with various microprocessors, microcomputers, and larger computers. This part combines three unique features: a complete time-of-day clock with alarm and one hundred year calendar, a programmable periodic interrupt and square-wave generator, and 50 bytes of low-power static RAM. The MC146818 uses high-speed CMOS technology to interface with 1 MHz processor buses, while consuming very little power.

The Real-Time Clock plus RAM has two distinct uses. First, it is designed as a battery-powered CMOS part (in an otherwise NMOS/TTL system) including all the common battery backed-up functions such as RAM, time, and calendar. Secondly, the MC146818 may be used with a CMOS microprocessor to relieve the software of the timekeeping workload and to extend the available RAM of an MPU such as the MC146805E2.

- Low-Power, High-Speed, High-Density CMOS
- Internal Time Base and Oscillator
- Counts Seconds, Minutes, and Hours of the Day
- Counts Days of the Week, Date, Month, and Year
- 3 V to 6 V Operation
- Time Base Input Options: 4.194304 MHz, 1.048576 MHz, or 32.768 kHz
- Time Base Oscillator for Parallel Resonant Crystals
- 40 to 200 µW Typical Operating Power at Low Frequency Time Base
- 4.0 to 20 mW Typical Operating Power at High Frequency Time Base
- Binary or BCD Representation of Time, Calendar, and Alarm
- 12- or 24-Hour Clock with AM and PM in 12-Hour Mode
- Daylight Savings Time Option
- Automatic End of Month Recognition
- Automatic Leap Year Compensation
- Microprocessor Bus Compatible
- MOTEL Circuit for Bus Universality
- Multiplexed Bus for Pin Efficiency
- Interfaced with Software as 64 RAM Locations
- 14 Bytes of Clock and Control Registers
- 50 Bytes of General Purpose RAM
- Status Bit Indicates Data Integrity
- Bus Compatible Interrupt Signals ($\overline{IRQ}$)
- Three Interrupts are Separately Software Maskable and Testable
  - Time-of-Day Alarm, Once-per-Second to Once-per-Day
  - Periodic Rates from 30.5 µs to 500 ms
  - End-of-Clock Update Cycle
- Programmable Square-Wave Output Signal
- Clock Output May Be Used as Microprocessor Clock Input
  - At Time Base Frequency ÷ 1 or ÷ 4
- 24-Pin Dual-In-Line Package

This document contains information on a new product. Specifications and information herein are subject to change without notice.